Linear Model Selection and Regularization

- Recall the linear model

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

- In the lectures that follow, we consider some approaches for extending the linear model framework. In the lectures covering Chapter 7 of the text, we generalize the linear model in order to accommodate *non-linear*, but still *additive*, relationships.
- In the lectures covering Chapter 8 we consider even more general *non-linear* models.

In praise of linear models!

- Despite its simplicity, the linear model has distinct advantages in terms of its *interpretability* and often shows good *predictive performance*.
- Hence we discuss in this lecture some ways in which the simple linear model can be improved, by replacing ordinary least squares fitting with some alternative fitting procedures.

Why consider alternatives to least squares?

- *Prediction Accuracy:* especially when $p > n$, to control the variance.
- *Model Interpretability:* By removing irrelevant features — that is, by setting the corresponding coefficient estimates to zero — we can obtain a model that is more easily interpreted. We will present some approaches for automatically performing *feature selection*.

Three classes of methods

- *Subset Selection*. We identify a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.
- *Shrinkage*. We fit a model involving all p predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as *regularization*) has the effect of reducing variance and can also perform variable selection.
- *Dimension Reduction*. We project the p predictors into a M-dimensional subspace, where $M < p$. This is achieved by computing M different *linear combinations*, or *projections*, of the variables. Then these M projections are used as predictors to fit a linear regression model by least squares.

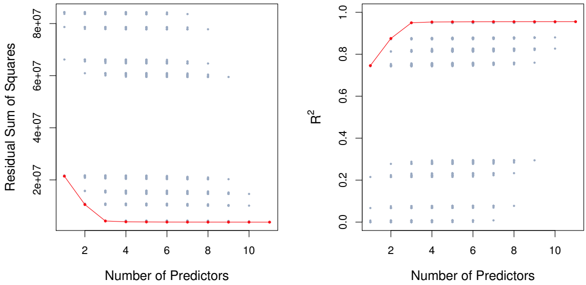
Subset Selection

Best subset and stepwise model selection procedures

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.

Example- Credit data set



For each possible model containing a subset of the ten predictors in the `Credit` *data set, the RSS and* R^2 *are displayed. The red frontier tracks the* best *model for a given number of predictors, according to RSS and* R^2*. Though the data set contains only ten predictors, the* x*-axis ranges from* 1 *to* 11*, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables*

Extensions to other models

- Although we have presented best subset selection here for least squares regression, the same ideas apply to other types of models, such as logistic regression.
- The *deviance*— negative two times the maximized log-likelihood— plays the role of RSS for a broader class of models.

Stepwise Selection

- For computational reasons, best subset selection cannot be applied with very large p. *Why not?*
- Best subset selection may also suffer from statistical problems when p is large: larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.
- Thus an enormous search space can lead to *overfitting* and high variance of the coefficient estimates.
- For both of these reasons, *stepwise* methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Forward Stepwise Selection

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest *additional* improvement to the fit is added to the model.

In Detail

Forward Stepwise Selection

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p - 1$:
    2.1 Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
    2.2 Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

More on Forward Stepwise Selection

- Computational advantage over best subset selection is clear.
- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors. *Why not? Give an example.*

Credit data example

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income` `student`, `limit`	`rating`, `income`, `student`, `limit`

The first four selected models for best subset selection and forward stepwise selection on the `Credit` *data set. The first three models are identical but the fourth models differ.*

Backward Stepwise Selection

- Like forward stepwise selection, *backward stepwise selection* provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

More on Backward Stepwise Selection

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection
- Like forward stepwise selection, backward stepwise selection is not guaranteed to yield the *best* model containing a subset of the p predictors.
- Backward selection requires that the *number of samples n is larger than the number of variables p* (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

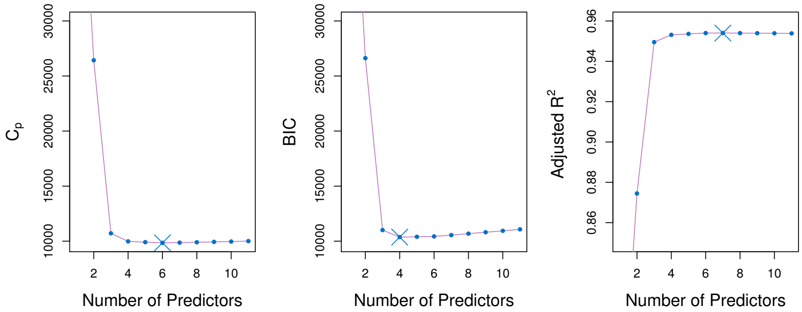
Choosing the Optimal Model

- The model containing all of the predictors will always have the smallest RSS and the largest R^2, since these quantities are related to the training error.
- We wish to choose a model with low test error, not a model with low training error. Recall that training error is usually a poor estimate of test error.
- Therefore, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors.

C_p, AIC, BIC, and Adjusted R^2

- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.
- The next figure displays C_p, BIC, and adjusted R^2 for the best model of each size produced by best subset selection on the `Credit` data set.

Credit data example



Now for some details

- *Mallow's C_p*:
$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right),$$
where d is the total # of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement.
- The *AIC* criterion is defined for a large class of models fit by maximum likelihood:
$$\mathrm{AIC} = -2\log L + 2 \cdot d$$
where L is the maximized value of the likelihood function for the estimated model.
- In the case of the linear model with Gaussian errors, maximum likelihood and least squares are the same thing, and C_p and AIC are equivalent. *Prove this.*

Details on BIC

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

- Like C_p, the BIC will tend to take on a small value for a model with a low test error, and so generally we select the model that has the lowest BIC value.
- Notice that BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where n is the number of observations.
- Since $\log n > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than C_p. See Figure on slide 19.

Adjusted R^2

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

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

  where TSS is the total sum of squares.
- Unlike C_p, AIC, and BIC, for which a *small* value indicates a model with a low test error, a *large* value of adjusted R^2 indicates a model with a small test error.
- Maximizing the adjusted R^2 is equivalent to minimizing $\frac{\text{RSS}}{n-d-1}$. While RSS always decreases as the number of variables in the model increases, $\frac{\text{RSS}}{n-d-1}$ may increase or decrease, due to the presence of d in the denominator.
- Unlike the R^2 statistic, the adjusted R^2 statistic *pays a price* for the inclusion of unnecessary variables in the model. See Figure on slide 19.

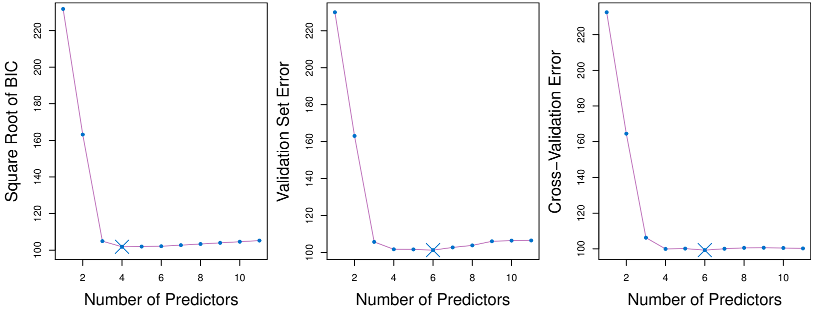
Validation and Cross-Validation

- Each of the procedures returns a sequence of models $\mathcal{M}_k$ indexed by model size $k = 0, 1, 2, \ldots$. Our job here is to select $\hat{k}$. Once selected, we will return model $\mathcal{M}_{\hat{k}}$

Validation and Cross-Validation

- Each of the procedures returns a sequence of models $\mathcal{M}_k$ indexed by model size $k = 0, 1, 2, \ldots$. Our job here is to select $\hat{k}$. Once selected, we will return model $\mathcal{M}_{\hat{k}}$
- We compute the validation set error or the cross-validation error for each model $\mathcal{M}_k$ under consideration, and then select the k for which the resulting estimated test error is smallest.
- This procedure has an advantage relative to AIC, BIC, C_p, and adjusted R^2, in that it provides a direct estimate of the test error, and *doesn't require an estimate of the error variance* σ^2.
- It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g. the number of predictors in the model) or hard to estimate the error variance σ^2.

Credit data example



Regularization

- When the number of observations or training examples m is not large enough compared to the number of feature variables n, over-fitting may occur.
- Tends to occur when large weights are found in x.
- What can we do to prevent over-fitting?
- Use L2-regularization
- Regularization :
- Minimize : (Loss Function) + (regularization term)

Shrinkage Methods

Ridge regression and *Lasso*

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

L2-Regularization

- Regularization term : $\lambda \|x\|_2^2$
  - $\lambda > 0$ is the regularization parameter
- For LSP, this becomes
  - Minimize $\|Ax - y\|^2 + \|Fx\|_2^2$
    - Regularization term restricts large value components
    - Special case of Tikhonov regularization
    - Can be computed directly ($O(n^3)$)
    - Or can use iterative methods (*e.g.* conjugate gradients method)
- For LRP, this becomes
  - Minimize $l_{avg}(v, x) + \lambda \|x\|_2^2$
  - Smooth and convex, can be solved using gradient descent, steepest descent, Newton, quasi-Newton, truncated Newton, CG methods

Ridge regression

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

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

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

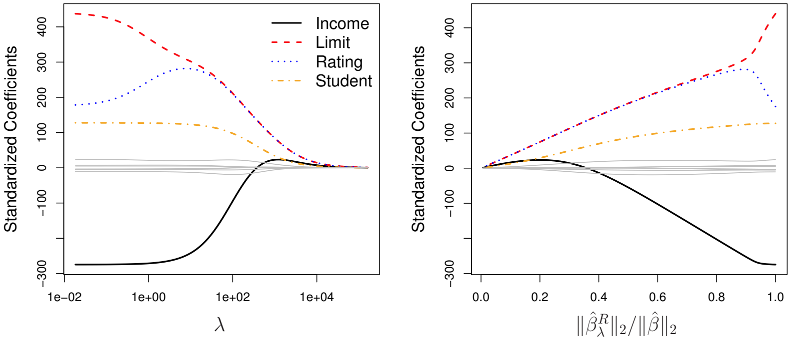
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2,$$

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

Ridge regression: continued

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda \sum_j \beta_j^2$, called a *shrinkage penalty*, is small when $\beta_1, \ldots, \beta_p$ are close to zero, and so it has the effect of *shrinking* the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.

Credit data example



Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.
- The notation $\|\beta\|_2$ denotes the ℓ_2 norm (pronounced "ell 2") of a vector, and is defined as $\|\beta\|_2 = \sqrt{\sum_{j=1}^p {\beta_j}^2}$.

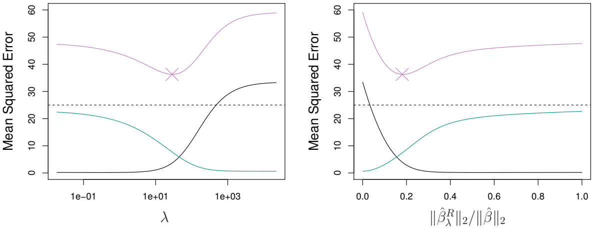
Ridge regression: scaling of predictors

- The standard least squares coefficient estimates are *scale equivariant*: multiplying X_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the jth predictor is scaled, $X_j\hat{\beta}_j$ will remain the same.
- In contrast, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.
- Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

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

Why Does Ridge Regression Improve Over Least Squares?

The Bias-Variance tradeoff



Simulated data with $n = 50$ observations, $p = 45$ predictors, all having nonzero coefficients. Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of λ and $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

The Lasso

- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model
- The *Lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the quantity

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

- In statistical parlance, the lasso uses an ℓ_1 (pronounced "ell 1") penalty instead of an ℓ_2 penalty. The ℓ_1 norm of a coefficient vector β is given by $\|\beta\|_1 = \sum|\beta_j|$.

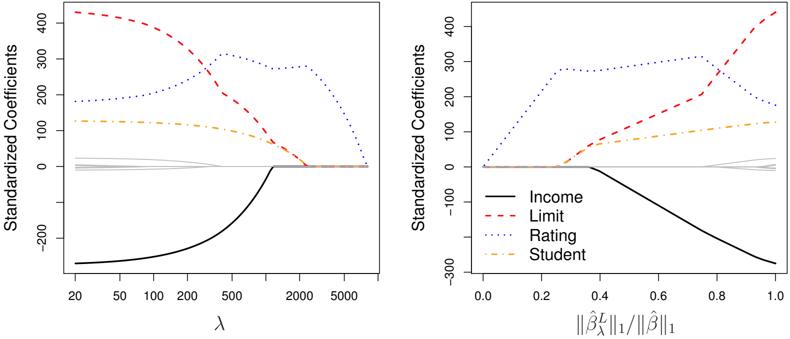
L1-Regularization

- Regularization term : $\lambda\|x\|_1$
- LSP : $\|Ax-y\|^2+\|Fx\|_2^2+\lambda\|x\|_1$
- LRP : $l_{avg}(v,x)+\lambda\|x\|_1$
- The regularization term penalizes all factors equally
- This makes the *x* **SPARSE**
  - A sparse *x* means reduced complexity
  - Can be viewed as a selection of relevant/important features
- Non-differentiable -> harder problem
  - Can transform into convex quadratic problem
    - minimize $\|Ax-y\|^2+\|Fx\|_2^2+\lambda\sum_{i=1}^{n}u_i$
    - subject to $-u_i \le x_i \le u_i,\ \ i=1,\dots,n$
  - and use standard convex optimization methods to solve, but these usually cannot handle large practical problems

The Lasso: continued

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- However, in the case of the lasso, the ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Hence, much like best subset selection, the lasso performs *variable selection*.
- We say that the lasso yields *sparse* models — that is, models that involve only a subset of the variables.
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Example: Credit dataset



The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

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

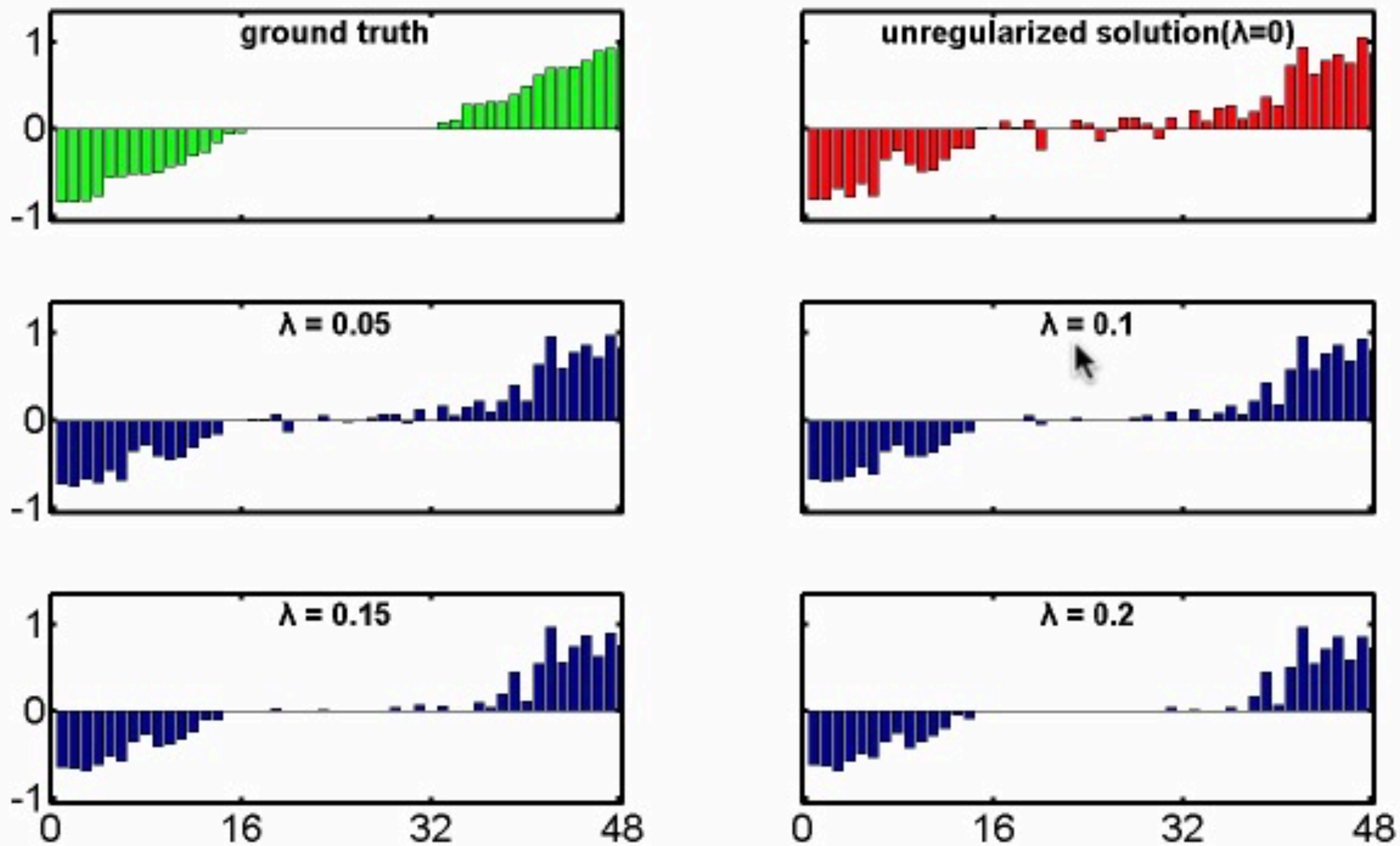
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \le s$$

and

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

respectively.

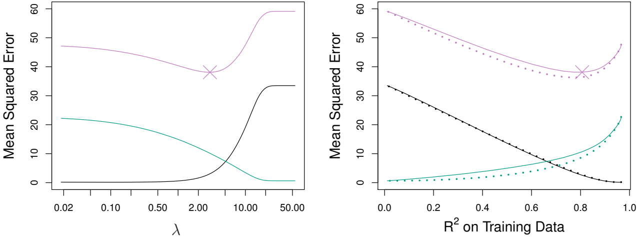
Effects of L1-Regularization



The limitations of the lasso

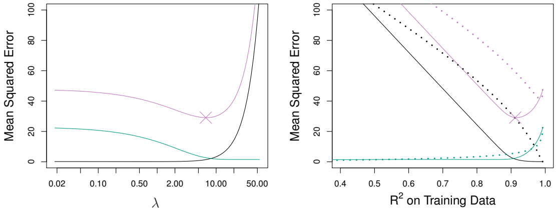
- If $p > n$, the lasso selects at most n variables. The number of selected genes is bounded by the number of samples.
- Grouped variables: the lasso fails to do grouped selection. It tends to select one variable from a group and ignore the others.

Comparing the Lasso and Ridge Regression



Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on simulated data set of Slide 32.* *Right:* *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their R^2 on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

Comparing the Lasso and Ridge Regression: continued



Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Slide 38, except that now only two predictors are related to the response.* *Right:* *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their* R^2 *on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

Example: murder data

Example: we study the murder rate (per 100K people) of $n = 2215$ communities in the U.S.[2] We have $p = 101$ attributes measured each community, such as

```
[1] "racePctHisp        "agePct12t21"        "agePct12t29"
[4] "agePct16t24"       "agePct65up"         "numbUrban"
[7] "pctUrban"          "medIncome"          "pctWWage"
...
```

Our goal is to predict the murder rate as a linear function of these attributes. For the purposes of interpretation, it would be helpful to have a linear model involving only a small subset of these attributes. (Note: interpretation here is *not causal*)

[2] Data from UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized`

With ridge regression, regardless of the choice of $\lambda < \infty$, we never get zero coefficient estimates. For $\lambda = 25,000$, which corresponds to approximately 5 degrees of freedom, we get estimates:

```
  racePctHisp         agePct12t21         agePct12t29
 0.0841354923        0.0076226029        0.2992145264
  agePct16t24         agePct65up            numbUrban
-0.2803165408        0.0115873137        0.0154487020
     pctUrban           medIncome             pctWWage
-0.0155148961       -0.0105604035       -0.0228670567
...
```

With the lasso, for about the same degrees of freedom, we get:

```
  agePct12t29         agePct16t24   NumKidsBornNeverMar
    0.7113530          -1.8185387            -0.6835089
PctImmigRec10       OwnOccLowQuart
    1.3825129           1.0234245
```

and all other coefficient estimates are zero. That is, we get exactly 5 nonzero coeffcients out of $p = 101$ total

Summary

* L2-Regression suppresses over-fitting
* L2-Regression does not add too much complexity to existing problems -> easy to calculate
* L1-Regression creates sparse answers, and better approximations in relevant cases
* L1-Regression problems are not differentiable -> need other ways of solving problem (using convex optimization techniques, iterative approaches, etc.)

Conclusions

- These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other.
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.
- However, the number of predictors that is related to the response is never known *a priori* for real data sets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular data set.

Selecting the Tuning Parameter for Ridge Regression and Lasso

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.
- *Cross-validation* provides a simple way to tackle this problem. We choose a grid of λ values, and compute the cross-validation error rate for each value of λ.
- We then select the tuning parameter value for which the cross-validation error is smallest.
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

“Elastic net” regression

Regression “elastic net” solves the following problem:

$$\hat{\beta} = \arg_{b \in \mathbb{R}^{p+1}} \min \left[\frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^T b)^2 + \lambda P_\alpha(b_1, \ldots, b_p) \right]$$

where

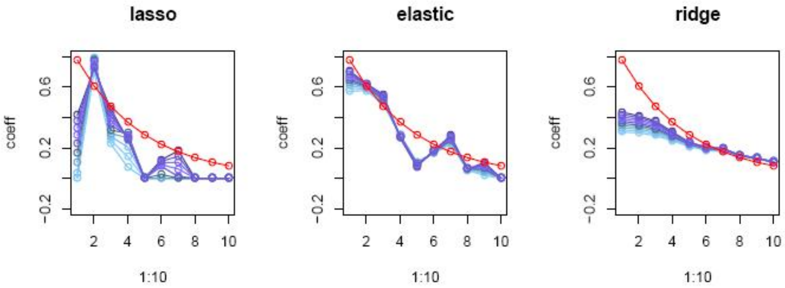
$$P_\alpha = \sum_{i=1}^{p} \left[\frac{1}{2}(1-\alpha) b_j^2 + \alpha |b_j| \right]$$

For different α we can have the following types of regression:

- $\alpha = 1$ lasso
- $\alpha = 0$ ridge
- $\alpha \in (0, 1)$ the general case of the elastic net

“Elastic net” regression

Ridge regression is known to shrink the coefficients of correlated predictors towards each other. Lasso is somewhat indifferent to very correlated predictors and will tend to pick one and ignore the rest.



Median MSE for the simulated examples

Method	Ex.1	Ex.2	Ex.3	Ex.4
Ridge	4.49 (0.46)	2.84 (0.27)	39.5 (1.80)	64.5 (4.78)
Lasso	3.06 (0.31)	3.87 (0.38)	65.0 (2.82)	46.6 (3.96)
Elastic Net	2.51 (0.29)	3.16 (0.27)	56.6 (1.75)	34.5 (1.64)
No re-scaling	5.70 (0.41)	2.73 (0.23)	41.0 (2.13)	45.9 (3.72)

Variable selection results

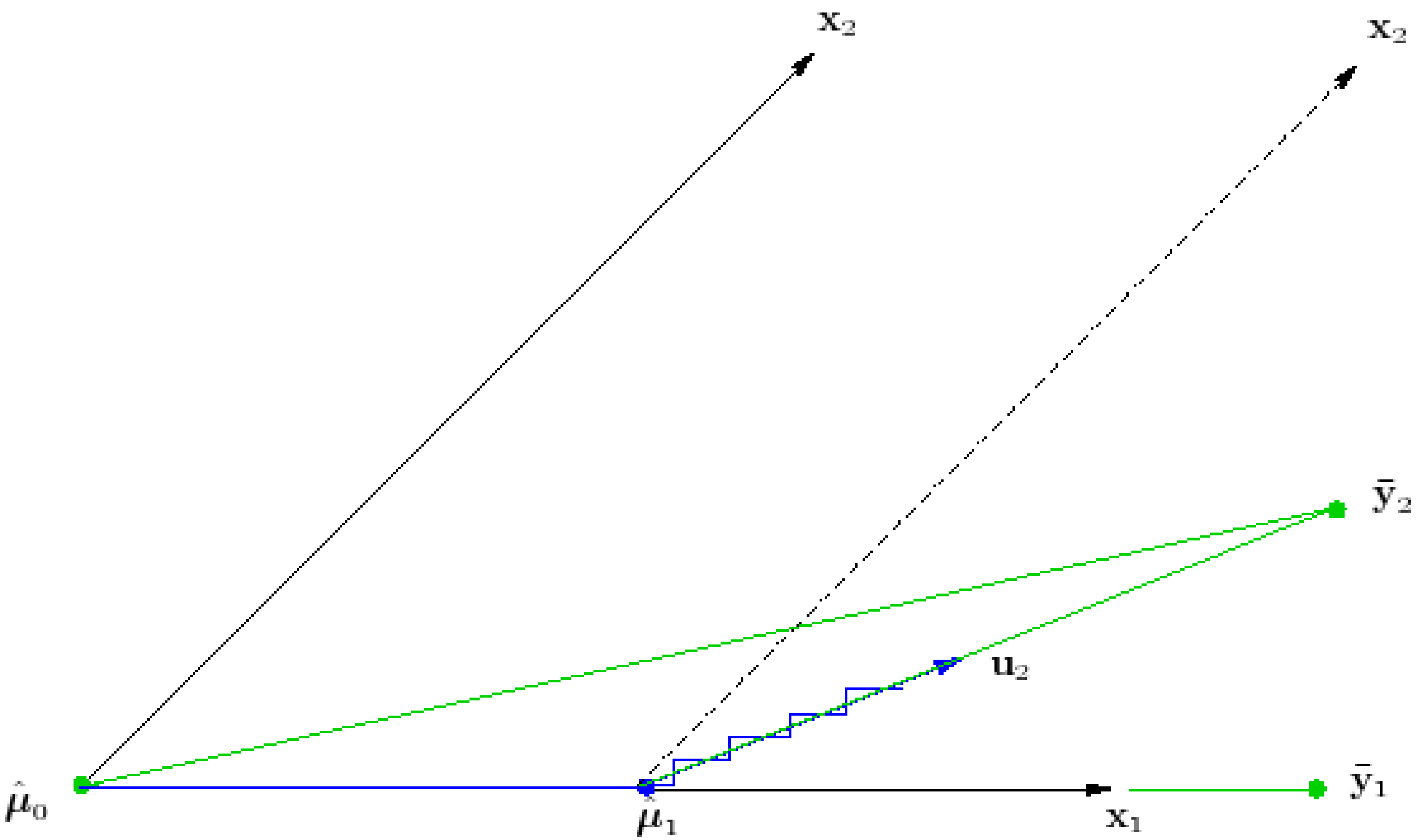
Method	Ex.1	Ex.2	Ex.3	Ex.4
Lasso	5	6	24	11
Elastic Net	6	7	27	16

LARS

- Least Angle Regression
  - Start with empty set
  - Select $\boldsymbol{x}_j$ that is most correlated with residuals $\boldsymbol{y} - \hat{\boldsymbol{\mu}}$
  - Proceed in the direction of $\boldsymbol{x}_j$ until another variable $\boldsymbol{x}_k$ is equally correlated with residuals
  - Choose equiangular direction between $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$
  - Proceed until third variable enters the active set, etc

- Step is always shorter than in OLS

Geometrical presentation



Computing LARS

- Every step a new variable enters the active set $\Rightarrow$ no more steps than variables
- New dicrection can be solved with linear algebra
- Step length by iterating over all variables not in active set
- By cleverly updating estimates from previous iteration, the computation cost will be comparable to OLS

Lasso modification

- LARS can be modified to give Lasso solution
- In the Lasso algorithm signs of the β_j and c_j must agree
- Take only as long LARS step as possible without changing the sign
- This works, if only one variable at a time enters the active set
- Unlike LARS, variables can be removed from active set