Week 4: Resampling methods and model selection

Slava Mikhaylov

POLS3003 Data Science and Big Data Analytics

Week 4 Outline

- Cross-validation
  - Validation-set approach
  - K-fold Cross-validation
- Bootstrap
  - Bootstrap mechanism
  - Bootstrap and prediction
  - The Bootstrap versus Permutation tests
- Linear Model Selection and Regularization
- Subset Selection
- Stepwise Selection
- Choosing the Optimal Model
- Shrinkage Methods
- Dimension Reduction Methods

Cross-validation

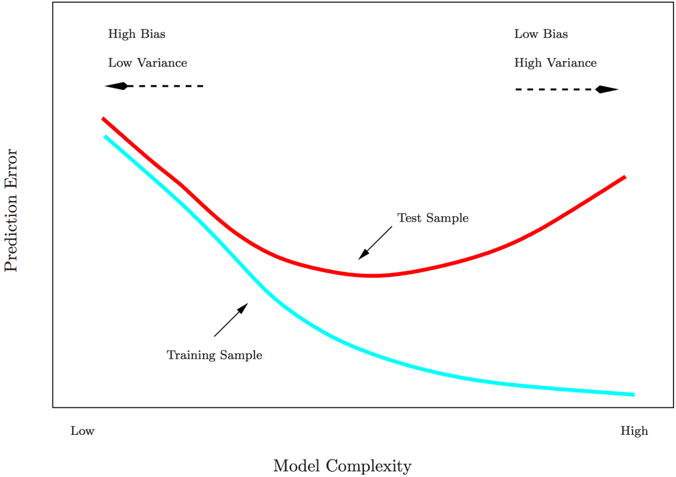
Resampling

- Today we discuss two resampling methods: cross-validation and the bootstrap.
- These methods refit a model of interest to samples formed from the training set, in order to obtain additional information about the fitted model.
- E.g., they provide estimates of test-set prediction error, and the standard deviation and bias of our parameter estimates.

Training Error versus Test error

- The test error is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
- In contrast, the training error can be easily calculated by applying the statistical learning method to the observations used in its training.
- Training error rate often is quite different from the test error rate, and in particular the former can dramatically underestimate the latter.

Training- versus Test-Set Performance



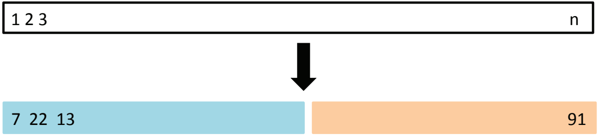
More on prediction-error estimates

- Best solution: a large designated test set. Often not available.
- Some methods make a mathematical adjustment to the training error rate in order to estimate the test error rate. These include the Cp statistic, AIC and BIC. They are discussed elsewhere in this course.
- Here we instead consider a class of methods that estimate the test error by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations.

Validation-set approach

- We randomly divide the available set of samples into two parts: a training set and a validation or hold-out set.
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate for qualitative response models.

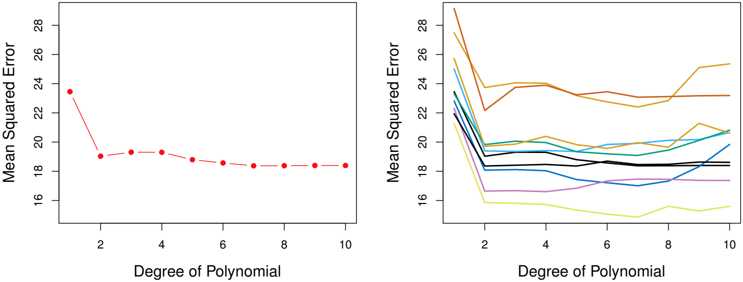
The Validation process



A random splitting into two halves: left part is training set, right part is validation set.

Example

- We want to compare linear vs higher-order polynomial terms in a linear regression with our Auto dataset.
- We randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.



Left panel shows single split; right panel shows multiple splits.

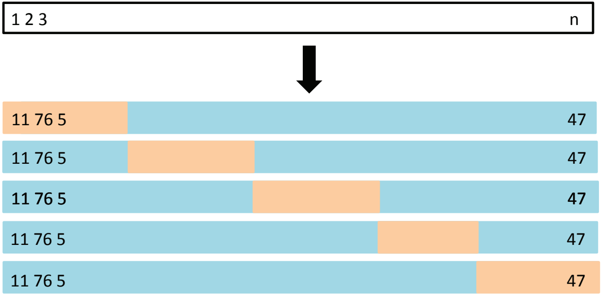
Drawbacks of validation set approach

- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.
- In the validation approach, only a subset of the observations – those that are included in the training set rather than in the validation set – are used to fit the model.
- This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire dataset. Why?

K-fold Cross-validation

- Very popular approach for estimating test error.
- Estimates can be used to select best model, and to give an idea of the test error of the final chosen model.
- Idea is to randomly divide the data into K equal-sized parts. We leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictions for the left-out kth part.
- This is done in turn for each part $k = 1, 2, \ldots, K$, and then the results are combined.

5-fold CV



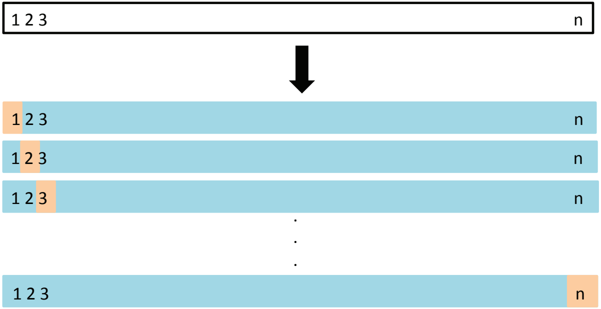
Mechanism

- Let the K parts be $C_1, C_2, \ldots, CK$, where CK denotes the indices of the observations in part k. There are n_k observations in part k: if N is a multiple of K, then $n_k = n/K$.
- Compute

$$CV_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$$

  where $\mathrm{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$, and $\hat{y}_i$) is the fit for observation i, obtained from the data with part k removed.
- Setting $K = n$ yields n-fold or leave-one out cross-validation (LOOCV).

LOOCV



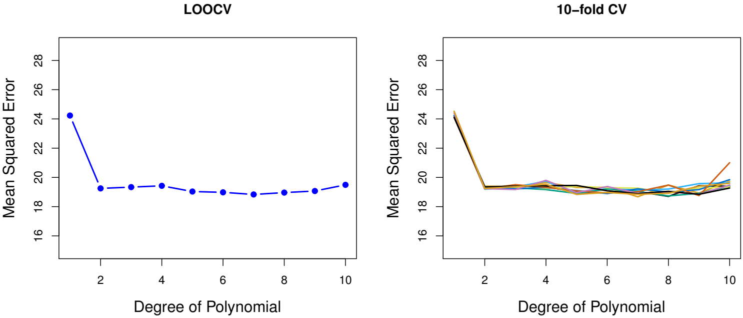
Special case of linear regression

- With least-squares linear or polynomial regression, there is a shortcut making the cost of LOOCV the same as that of a single model fit.
- The following formula holds:

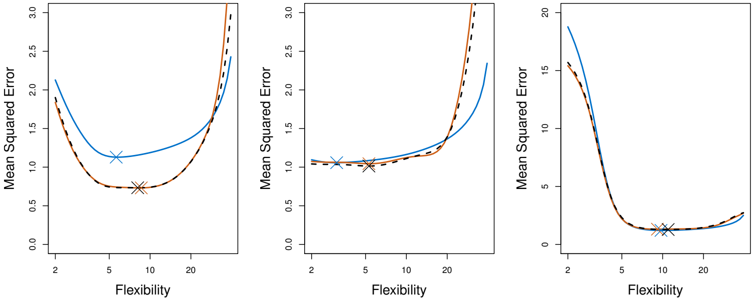
$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2$$

  where $\hat{y}_i$ is the ith fitted value from the original least squares fit, and h_i is the leverage (diagonal of the "hat" matrix). This is similar to the ordinary MSE, except the ith residual is divided by $(1 - h_i)$.
- The estimates from each fold are highly correlated and hence their average can have high variance.
- A better choice is $K = 5$ or 10.

Auto data example



True and estimated test MSE for the simulated data



Additional issues with CV

- Since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward.
- This bias is minimized when $K = n$ (LOOCV), but this estimate has high variance, as noted earlier.
- $K = 5$ or 10 provides a good balance for this bias-variance tradeoff.

CV for classification

- We divide the data into K roughly equal-sized parts $C_1, C_2, \ldots, C_K$. C_k denotes the indices of the observations in part k. There are n_k observations in part k: if n is a multiple of K, then $n_k = n/K$.
- Compute

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err_k}$$

where $\mathrm{Err_k} = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$.

- The estimated standard deviation of CV_K is

$$\widehat{\mathrm{SE}}(CV_K) = \sqrt{\sum_{k=1}^{K} (\mathrm{Err}_k - \mathrm{E\bar{r}r}_k)^2/(K-1)}$$

- This is a useful estimate, but strictly speaking, not quite valid. Why not?

CV: right and wrong

- Consider a simple classifier applied to some two-class data:
  1. Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels.
  2. We then apply a classifier such as logistic regression, using only these 100 predictors.
- How do we estimate the test set performance of this classifier?
- Can we apply cross-validation in step 2, ignoring step 1?

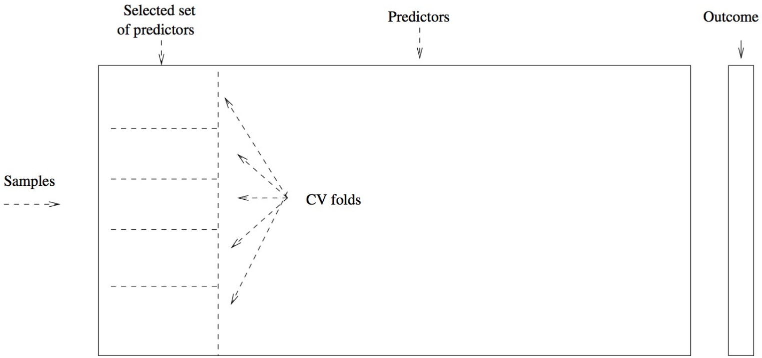
CV: right and wrong

- This would ignore the fact that in Step 1, the procedure has already seen the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.
- It is easy to simulate realistic data with the class labels independent of the outcome, so that true test error =50%, but the CV error estimate that ignores Step 1 is zero. (You can try doing this in class later today.)

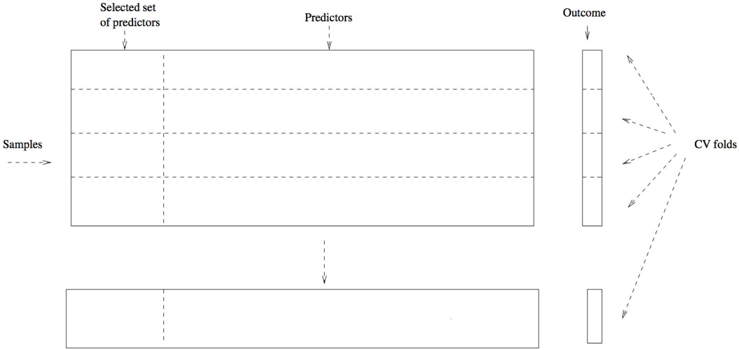
CV: right and wrong

- Incorrect: Apply cross-validation in step 2.
- Correct: Apply cross-validation to steps 1 and 2.

Wrong


Selected set of predictors
Predictors
Outcome
Samples
CV folds

Right


Selected set of predictors
Predictors
Outcome
Samples
CV folds

Bootstrap

Bootstrap

- The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- E.g., it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

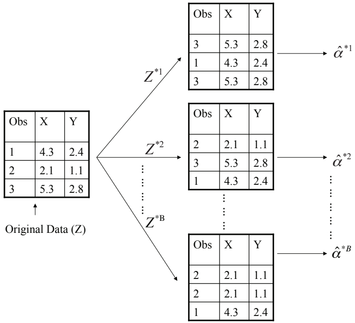
Where does the name came from?

- The use of the term bootstrap derives from the phrase to pull oneself up by one's bootstraps, widely thought to be based on one of the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:
  The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.
- It is not the same as the term bootstrap used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.

Intuition

- We usually care about uncertainty around our estimates from a sample. One way would be to repeatedly sample the population. But we cannot do that in real world.
- The bootstrap approach allows us to use replicate this process of obtaining new data sets, so that we can estimate the variability of our estimate without generating additional samples.
- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set with replacement.
- Each of these “bootstrap data sets” is created by sampling with replacement, and is the same size as our original dataset. As a result some observations may appear more than once in a given bootstrap data set and some not at all.

Example with three observations



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

- A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations.
- Each bootstrap dataset contains n observations, sampled with replacement from the original data set.
- Each bootstrap data set is used to obtain an estimate of α.

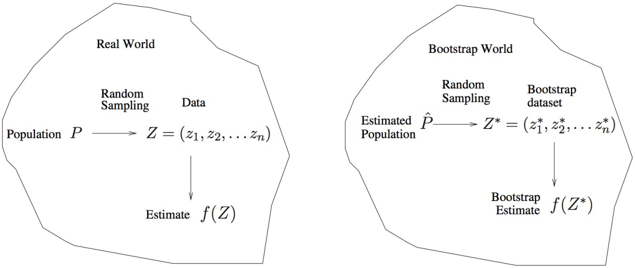
Mechanism

- Denoting the first bootstrap dataset by Z^{*1}, we use Z^{*1} to produce a new bootstrap estimate for α, which we call $\hat{\alpha}^{*1}$.
- This procedure is repeated B times for some large value of B (say 1000 or 10,000), in order to produce B different bootstrap datasets, $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$, and B corresponding α estimates, $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$.
- We estimate the standard error of these bootstrap estimates using the formula

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^{B} (\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^{*})^2}$$

- This serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set.

A general picture for the bootstrap


Real World
Random Sampling
Data
Population P
$Z = (z_1, z_2, \dots z_n)$
Estimate $f(Z)$
Bootstrap World
Random Sampling
Bootstrap dataset
Estimated Population $\hat{P}$
$Z^* = (z_1^*, z_2^*, \dots z_n^*)$
Bootstrap Estimate $f(Z^*)$

The bootstrap in general

- In more complex data situations, figuring out the appropriate way to generate bootstrap samples can require some thought.
- For example, if the data is a time series, we can't simply sample the observations with replacement (why not?).
- We can instead create blocks of consecutive observations, and sample those with replacements. Then we paste together sampled blocks to obtain a bootstrap dataset.

Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate (e.g. median).
- Also provides approximate confidence intervals for a population parameter. Also called a Bootstrap Percentile confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Bootstrap and prediction error

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: there is no overlap. This is crucial for its success.
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample.
- This will cause the bootstrap to seriously underestimate the true prediction error.
- The other way around – with original sample = training sample, bootstrap dataset = validation sample – is even worse.

Removing the overlap

- Can partly fix this problem by only using predictions for those observations that did not (by chance) occur in the current bootstrap sample.
- But the method gets complicated, and in the end, cross-validation provides a simpler, more attractive approach for estimating prediction error.

The Bootstrap versus Permutation tests

- The bootstrap samples from the estimated population, and uses the results to estimate standard errors and confidence intervals.
- Permutation methods sample from an estimated null distribution for the data, and use this to estimate p-values and False Discovery Rates for hypothesis tests.
- The bootstrap can be used to test a null hypothesis in simple situations. E.g. if $\theta = 0$ is the null hypothesis, we check whether the confidence interval for θ contains zero.
- Can also adapt the bootstrap to sample from a null distribution (see Efron and Tibshirani book "An Introduction to the Bootstrap" 1993, chapter 16) but there's no real advantage over permutations.

Linear Model Selection and Regularization

Linear Model Selection and Regularization

- Recall the linear model

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

- In the next weeks, we consider some approaches for extending the linear model framework.
- Next week, we generalize the linear model in order to accommodate non-linear, but still additive, relationships.
- The following week we consider even more general non-linear models.

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 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.

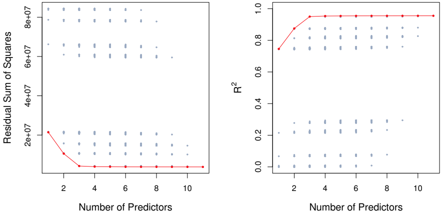
Subset Selection

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 $k = 1, 2, \ldots, p$:
   - Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   - 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



- 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 $k = 0, \ldots, p - 1$:
   - Consider all $p - k$ models that augments the predictors in $\mathcal{M}_k$ with one additional predictor.
   - Choose the best among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here best is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Backward Stepwise Selection

- 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.

Backward Stepwise Selection: details

Backward Stepwise Selection

1. Let $\mathcal{M}_p$ denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   - Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   - Choose the best among these k models, and call it $\mathcal{M}_{k-1}$. Here best is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

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.

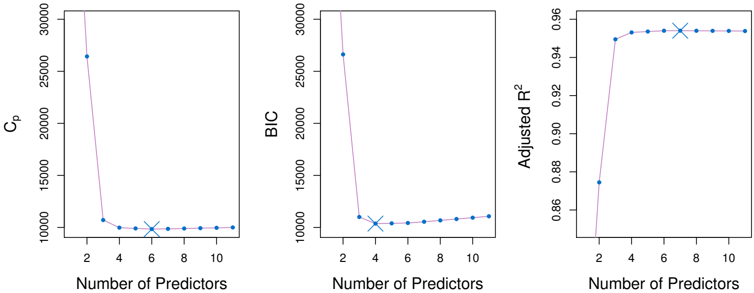
Estimating test error: two approaches

- We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting.
- We can directly estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in previous lectures.
- We illustrate both approaches next.

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.

Example: Credit data



Mallow's C_p

- Mallow's C_p:

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

  where d is the total number 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:

$$AIC = 2logL + 2\dot{d}$$

  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.

Details on BIC

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

- 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.

Adjusted R^2

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

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

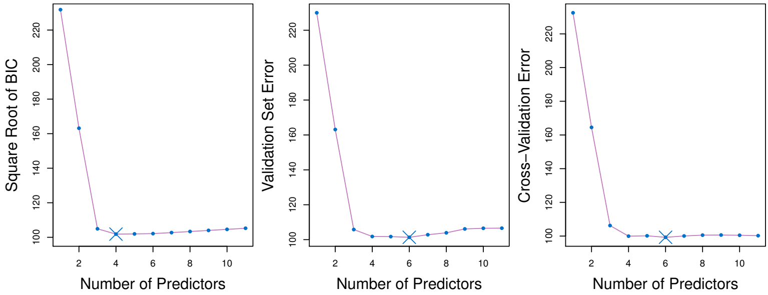
  where TSS is the total sum of squares.

- 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 $\frac{RSS}{n-d-1}$. While RSS always decreases as the number of variables in the model increases, $\frac{RSS}{n-d-1}$ 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.

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 R2, 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.

Example: Credit data



Explaining the example above

- The validation errors were calculated by randomly selecting three-quarters of the observations as the training set, and the remainder as the validation set.
- The cross-validation errors were computed using $k = 10$ folds. In this case, the validation and cross-validation methods both result in a six-variable model.
- However, all three approaches suggest that the four-, five-, and six-variable models are roughly equivalent in terms of their test errors.
- In this setting, we can select a model using the one-standard-error rule. We first calculate the standard error of the estimated test MSE for each model size, and then select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve.

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.

Ridge regression

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

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

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

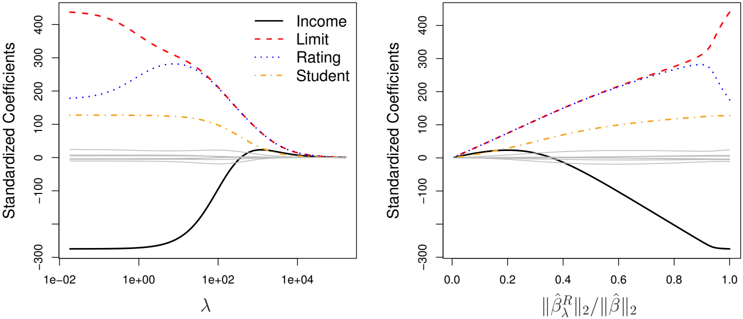
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2,$$

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

Example: Credit data



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 $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$, 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 $\|\hat{\beta}\|_2 = \sqrt{\sum_{j=1}^p \beta_j^2}$.

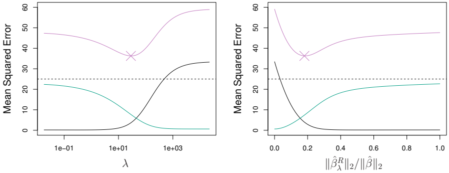
Ridge regression: scaling of predictors

- 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

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

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

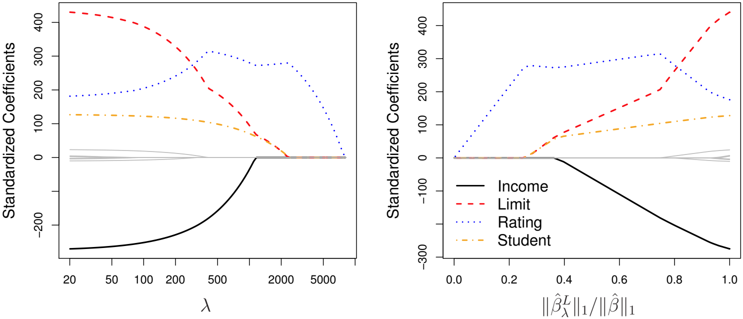
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

- 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 $\|\beta\|_1 = \sum|\beta_j|$.

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 data



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

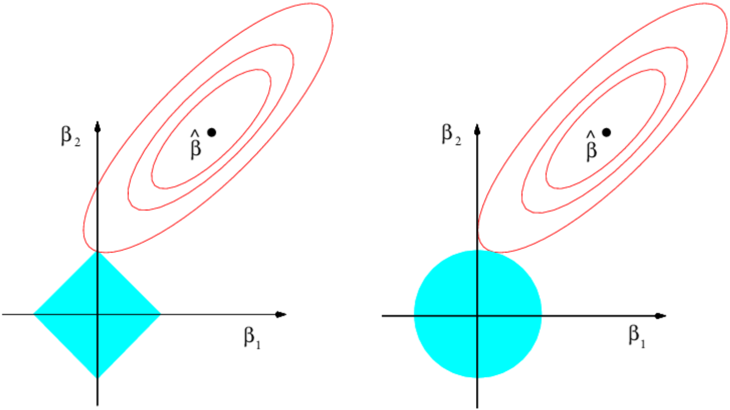
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \le s$$

and

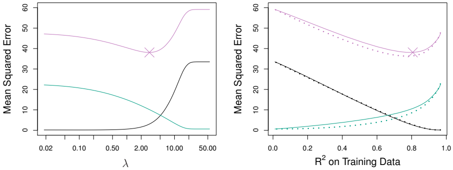
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \le s,$$

respectively.

The Lasso Picture

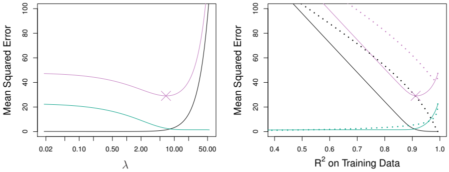


Comparing the Lasso and Ridge Regression



- Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on simulated data set. (slide 33)
- 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 previous slide, 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.

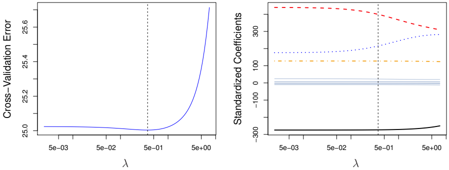
Take away message

- 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

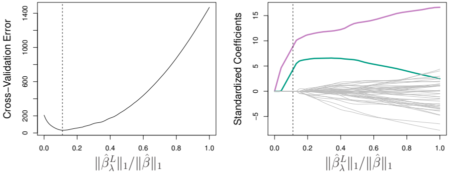
- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.
- Cross-validation provides a simple 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.

Example: Credit data



- Left: Cross-validation errors that result from applying ridge regression to the Credit data set with various values of λ.
- Right: The coefficient estimates as a function of λ. The vertical dashed lines indicates the value of λ selected by cross-validation.

Simulated data example



- Left: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from slide 40.
- Right: The corresponding lasso coefficient estimates are displayed.
- The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.

Dimension Reduction Methods

- The methods that we have discussed so far today have involved fitting linear regression models, via least squares or a shrunken approach, using the original predictors, $X_1, X_2, \ldots, X_p$.
- We now explore a class of approaches that transform the predictors and then fit a least squares model using the transformed variables.
- We will refer to these techniques as dimension reduction methods.

Dimension Reduction Methods: details

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

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

  for some constants $\phi_{m1}, \ldots, \phi_{mp}$.
- We can then fit the linear regression model,

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

  using ordinary least squares.
- Note that in the linear regression model above, the regression coefficients are given $\theta_0, \theta_1, \ldots, \theta_M$. If the constants $\phi_{m1}, \ldots, \phi_{mp}$ are chosen wisely, then such dimension reduction approaches can often outperform OLS regression.

- Notice that from the definition of Z_m above,

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

where

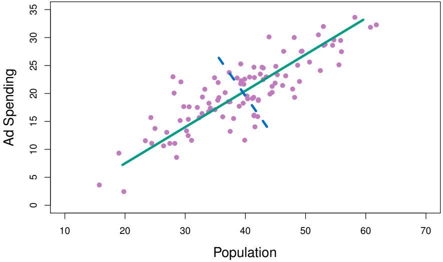
$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}.$$

- Hence linear regression model above can be thought of as a special case of the original linear regression model.
- Dimension reduction serves to constrain the estimated β_j coefficients, since now they must take the above form.
- Doing this we can win in the bias-variance tradeoff.

Principal Components Regression

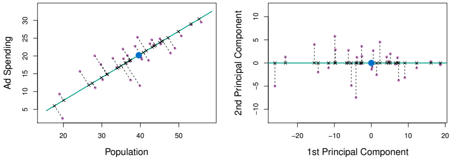
- Here we apply principal components analysis (PCA) (discussed later in the course) to define the linear combinations of the predictors, for use in our regression.
- The first principal component is that (normalized) linear combination of the variables with the largest variance.
- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.

Pictures of PCA



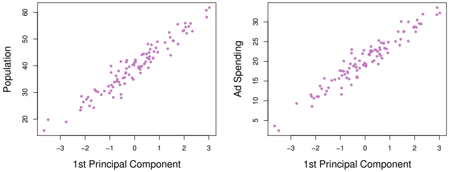
- The population size (*pop*) and ad spending (*ad*) for 100 different cities are shown as purple circles.
- The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.

Pictures of PCA: continued



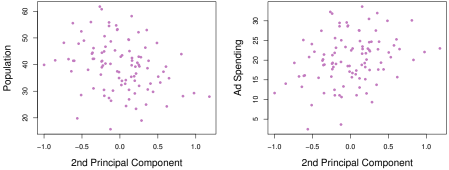
- A subset of the advertising data.
- Left: The first principal component, chosen to minimize the sum of the squared perpendicular distances to each point, is shown in green. These distances are represented using the black dashed line segments.
- Right: The left-hand panel has been rotated so that the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

Pictures of PCA: continued



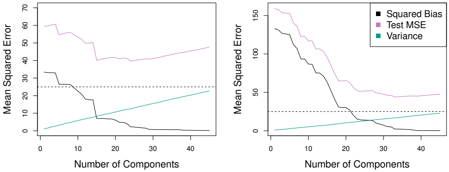
- Plots of the first principal component scores z_{i1} versus *pop* and *ad*.
- The relationships are strong.

Pictures of PCA: continued



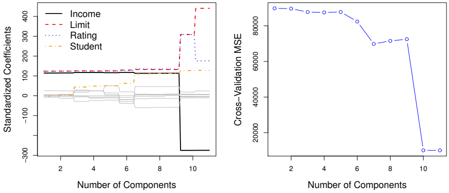
- Plots of the second principal component scores z_{i2} versus *pop* and *ad*.
- The relationships are weak.

Application of Principal Components Regression



- PCR was applied to two simulated data sets.
- The black, green, and purple lines correspond to squared bias, variance, and test mean squared error, respectively.
- Left: Simulated data from slide 33.
- Right: Simulated data from slide 40.

Choosing the number of directions M



- Left: PCR standardized coefficient estimates on the *Credit* data set for different values of M.
- Right: The 10-fold cross validation MSE obtained using PCR, as a function of M.

Partial Least Squares

- PCR identifies linear combinations, or directions, that best represent the predictors $X_1, \ldots, X_p$.
- These directions are identified in an unsupervised way, since the response Y is not used to help determine the principal component directions.
- That is, the response does not supervise the identification of the principal components.
- Consequently, PCR suffers from a potentially serious drawback:
  - there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.

Partial Least Squares: continued

- Like PCR, PLS is a dimension reduction method, which first identifies a new set of features $Z_1, \ldots, Z_M$ that are linear combinations of the original features, and then fits a linear model via OLS using these M new features.
- But unlike PCR, PLS identifies these new features in a supervised way – that is, it makes use of the response Y in order to identify new features that not only approximate the old features well, but also that are related to the response.
- Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.

Details of Partial Least Squares

- After standardizing the p predictors, PLS computes the first direction Z_1 by setting each ϕ_{1j} equal to the coefficient from the simple linear regression of Y onto X_j.
- One can show that this coefficient is proportional to the correlation between Y and X_j.
- Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.
- Subsequent directions are found by taking residuals and then repeating the above prescription.

Summary

- Model selection methods are an essential tool for data analysis, especially for big datasets involving many predictors.
- Research into methods that give sparsity, such as the lasso is an especially hot area.
- Later, we will return to sparsity in more detail, and will describe related approaches such as the elastic net.