Week 4: Model Assessment and Selection

- An Introduction to Statistical Learning: with Applications in R, J. Gareth, D. Witten, T. Hastie and R. Tibshirani (2021), New York : Springer.
  - 5.1 Cross-Validation
  - 6.1 Subset Selection
  - 6.2 Shrinkage Methods
- The elements of statistical learning: data mining, inference and predictions, T. Hasties, R. Tibshirani and J. Friedman (2001) Springer.
  - 3.3 Subset Selection
  - 3.4.1 Ridge Regression and 3.4.2 The Lasso
  - 7.1-7.7 and 7.10 Cross Validation

4.1 Model Assessment and Selection

Model Assessment and Selection

Today, we are going to discuss some of the most important concepts that arise in selecting a statistical model for a specific data set.

There are two separate goals that we might have in mind:

(a) Model selection: is estimating the performance of different models to choose the best one.

(b) Model assessment: having chosen a final model, estimating its test error on new data.

Measuring the Quality of Fit

One of the most commonly used measures of quality of fit is the mean squared error (MSE), given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ gives for the ith observation.

- This is the training MSE, which means that it is computed using the training data that was used to fit the model.
- We are more interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.
- In other words, we are interested in $\hat{f}(x_0) \approx y_0$, where (x_0, y_0) is a previously unseen test observation not used to train the model.
- If we had a large number of test observations we would compute the average:

$$\text{Ave}(\hat{f}(x_0) - y_0)^2.$$

Example: Polynomial fitting

Assume simulated data with $n = 20$ of the form $y_i = x_i + \epsilon_i$, where $\epsilon_i \sim N(0, 0.25^2)$. The following model is "correct" for every $k \geq 1$, if $\beta_0 = \beta_2 = \ldots = \beta_k = 0$ and $\beta_1 = 1$.

Let us now increase the level of flexibility and fit a polynomial of order 5, 10 and 15 to this simulated data. That is:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_k x_i^k + \epsilon_i.$$

We can see that

- with increasing order the polynomials fit the data more closely.
- the polynomials estimate the true f poorly because they are too wiggly

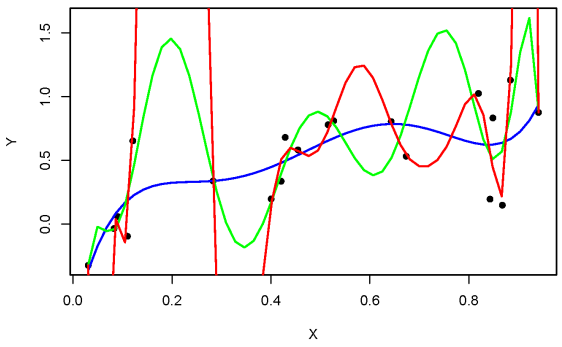


Figure 1: Fitted polynomial curves for polynomial of order $k = 5$ (blue), 10 (green) and 15 (red)

Cont. Model Assessment and Selection

If we know the true f, we can also compute the test MSE as a function of flexibility.

- The test MSE initially declines.
- However, at some point, the test MSE levels off and then starts to increase again.

The Figure 2 depicts the average training MSE (thick blue line) as a function of the model flexibility.

- The training MSE declines monotonically as flexibility increases.
- When a given model yields a small training MSE but a large test MSE, we are said to be overfitting the data.

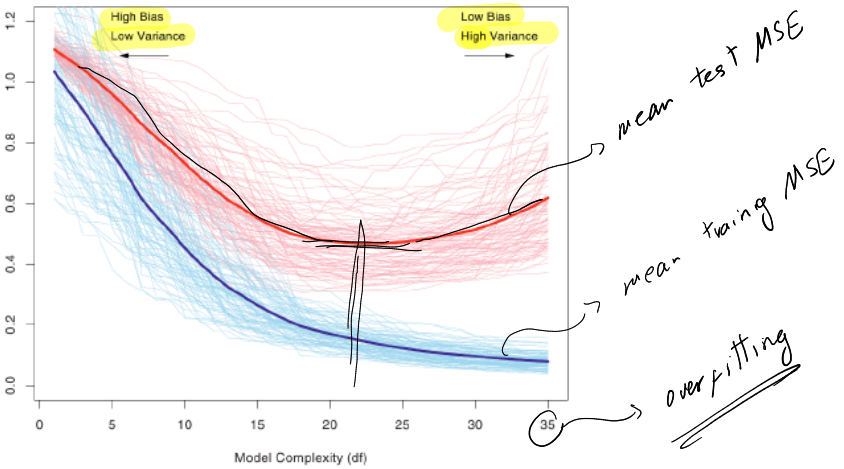


Figure 2: Training MSE (thin blue lines), test MSE (thin red lines) and their averages (thick lines).

The Bias-Variance Trade-Off

If we assume that $Y = f(X) + \varepsilon$, where $\boldsymbol{E}(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma_\varepsilon^2$, the expected test MSE, for a given value x_0, can be calculated as follows

$$\begin{aligned}\boldsymbol{E}(y_0 - \hat{f}(x_0))^2 &= \sigma_\varepsilon^2 + (\boldsymbol{E}(\hat{f}(x_0)) - f(x_0))^2 + \boldsymbol{E}(\hat{f}(x_0) - \boldsymbol{E}(\hat{f}(x_0)))^2 \\ &= \sigma_\varepsilon^2 + \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)).\end{aligned}$$

So, to minimise the expected test error we need to select a model that simultaneously achieves low variance and low bias.

- Variance: the amount by which $\hat{f}$ would change if we estimated it using a different training data set.
  - If a method has high variance, then small changes in the training data can result in substantial changes in $\hat{f}$.

$$E(y_0 - \hat{f}(x_0))^2 = E(y_0 \pm E(\hat{f}(x_0)) - \hat{f}(x_0) \pm f(x_0))^2$$

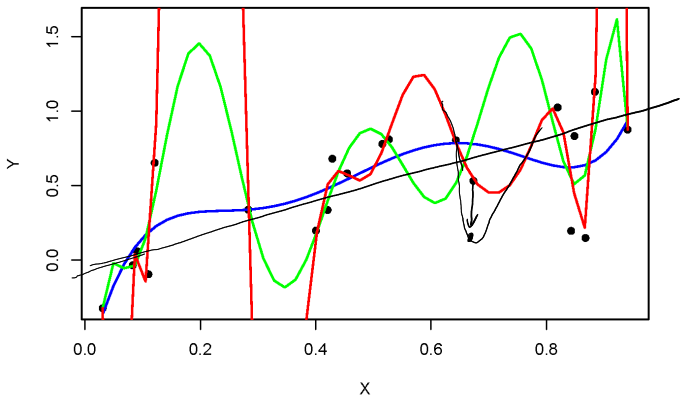
$$= E(\underbrace{y_0 - f(x_0)}_{\varepsilon_0})^2 + E\left[E(\hat{f}(x_0)) - f(x_0)\right]^2$$

$$+ E(\hat{f}(x_0) - E(\hat{f}(x_0)))^2$$

$$= \underbrace{\sigma_\varepsilon^2}_{\text{irreducible}} + \left[\text{bias}(\hat{f}(x_0))\right]^2 + \text{Var}(\hat{f}(x_0))$$

- Bias: the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.

Example: Cont. Polynomial fitting



We can see that

- the polynomial of order 15 is following the observations very closely.
- It has high variance because changing any one of these data points may cause the estimate $\hat{f}$ to vary considerably.
- On the other hand, the linear fit has small variance, because changing any single observation will likely cause only a tiny shift in the position of the line.

Statistical Decision Theory

- $X \in \mathbb{R}^p$: a real valued random input vector
- $Y \in \mathbb{R}$: a real valued random output variable
- $p(x, y)$: joint density function of X and Y
- $f(X)$: a function for predicting Y given values of X.

The theory requires a loss function $L(Y, f(X))$ for penalizing errors in prediction.

- **Squared error loss**: $L(y_i, f(x_i)) = (y_i - f(x_i))^2$
- **Deviance loss**: $L(y_i, f(x_i)) = -\ell(\beta; x_i, y_i)$ where β determines the regression function f.
- **0-1 loss**: In a classification problem (such as logistic regression), the response y_i is categorical, and $f(x_i) \in \{1, \ldots, K\}$. Then mis-classification loss is $L(y_i, f(x_i)) = \mathbf{1}_{\{y_i \neq f(x_i)\}}$.
- **Exponential loss**: Supposes $f(x_i) \in \mathbb{R}$ and $y_i \in \{-1, +1\}$, and $L(y_i, f(x_i)) = \exp(-y_i f(x_i))$.

$$L(y_i, f(x_i)) = \begin{cases} 1 & y_i \neq f(x_i) \\ 0 & y_i = f(x_i) \end{cases}$$

Cont. Statistical Decision Theory

Aim: Find ("learn") the function f which minimises

$$\boldsymbol{E}_{(X,Y)}[L(Y, f(X))] = \iint L(y, f(x))p(x, y)dxdy.$$

The marginal $p(x) = \int p(x, y)dy$ expresses the modelling focus: what regions of $\mathbb{R}^d$ are to be modelled well by f?

For instance,

- if predictions on an interval $[-1, +1]$ are equally important, then a valid assumption is that $p(x)$ is the uniform distribution on $[-1, +1]$;
- if prediction around 0 are more important, one may decide for $p(x) = \mathcal{N}(x; 0, 1)$.

Example: Square Error Loss and Regression

For the square error loss $L(Y, f(X)) = (Y - f(X))^2$, we need to minimize

$$\begin{aligned} E_{(X,Y)}(Y - f(X))^2 &= \int \int (y - f(x))^2 p(x, y) dx dy \\ &= \int \int (y - f(x))^2 p(y|x) p(x) dy dx \\ &= E_X E_{Y|X}([Y - f(X)]^2 | X). \end{aligned}$$

It suffices to minimize this criterion pointwise:

$$f(x) = \text{argmin}_c E_{Y|X}([Y - c]^2 | X = x).$$

The solution is

$$f(x) = E(Y|X = x)$$

the conditional expectation, also known as the regression function.

Definitions of Errors

Assume $\mathcal{T} = (x_1, y_1), \ldots, (x_N, y_N)$ is a training set drawn from $p(x, y)$, and a function $\hat{f}(x)$ has been fitted.

- The Training Error

$$\overline{\text{err}}(\mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

  quantifies the discrepancy between fitted value $\hat{y}_i = \hat{f}(x_i)$ and the response y_i.

- The Generalisation or test error is

$$\begin{aligned} \text{Err}_{\mathcal{T}} &= E_{(X,Y)|\mathcal{T}}[L(Y, \hat{f}(X))] \\ &= E[L(Y, \hat{f}(X))|\mathcal{T}] = \int L(y, \hat{f}(x)) p(x, y) dx dy \end{aligned}$$

  $\hat{f}$ is based on the training dataset

  is obtained by averaging the loss over all new data pairs, where the model fit based on $\mathcal{T}$ is left fixed.

  - Small $\text{Err}_{\mathcal{T}}$ is desirable since then $\hat{f}$ generalises well and yields small loss.
  - The generalisation error $\text{Err}_{\mathcal{T}}$ does depend on the training data.
  - The test error refers to the error for this specific training set.

Cont. Definitions of Errors

- The Expected error is defined as

$$\begin{aligned} \text{Err} &= E_{\mathcal{T}}[\text{Err}_{\mathcal{T}}] \\ &= \int \text{Err}_{\mathcal{T}}\, p(\mathcal{T}) d\mathcal{T} = E[L(Y, \hat{f}(X))] \end{aligned}$$

  averages the generalisation error over all training sets $\mathcal{T}$ of size N.

  - If the expected error is small, then the general approach taken in fitting $\hat{f}$ is right.
  - The quality of an individual $\hat{f}$ still depends on the training set.
  - In applications, one is interested in $\text{Err}_{\mathcal{T}}$, which describes the quality of the presently fitted model.
  - Estimates for $\text{Err}_{\mathcal{T}}$ can only be obtained by estimating Err.

Optimism of the Training Error Rate

- The in-sample error is defined as:

$$\begin{aligned}\text{Err}_{\text{in}} &= \frac{1}{N}\sum_{i=1}^{N} Err(x_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} E_{\mathbf{Y}}[L(Y_i, \hat{f}(x_i))|\mathcal{T}] \\ &= \frac{1}{N}\sum_{i=1}^{N} \int L(y, \hat{f}(x_i))p(y|x = x_i)dy\end{aligned}$$

  where the expected loss is calculated when the response is resampled at each location x_i, with $\hat{f}$ left unchanged.

  - This is effectively the generalisation error, conditional on the new data (X, Y) arriving at the old training coordinates $x_1, \ldots, x_N$, (we observe N new response values at each of the training points x_i).
  - The best guess for Err is given by $\text{Err}_{\text{in}}(\mathcal{T})$.

Cont. Optimism of the Training Error Rate

- $\hat{f}$ has been chosen to minimise the loss $\overline{\text{err}}$ at the training data.
  $\Rightarrow$ Resampling the data, but keeping $\hat{f}$ constant, will typically increase the loss.
- The difference
$$\text{op}(\mathcal{T}) = \text{Err}_{\text{in}}(\mathcal{T}) - \overline{\text{err}}(\mathcal{T})$$
is called the optimism of the training error.

The optimism depends on

- the fitted regression function $\hat{f}$,
- the responses y_i at x_i.

Taking the average over the responses y_i (drawn from $p(y|x = x_i)$) and thus the average over the possible $\hat{f}$ defines the average optimism $E_{\mathbf{y}}[\text{op}]$. The locations x_i are fixed according to $\mathcal{T}$ and the averages are taken over resampled y_i.

Cont. Optimism of the Training Error Rate

For most loss functions including "squared" and "0-1", the average optimism is:

$$E_{\mathbf{y}}[\text{op}] = \frac{2}{N}\sum_{i=1}^{N}\text{Cov}(\hat{y}_i, y_i) = \frac{2}{N}\text{tr}(\text{Cov}(\boldsymbol{H}\boldsymbol{y}, \boldsymbol{y})) = \frac{2\sigma^2}{N}\text{tr}(\boldsymbol{H})$$

We hence estimate:

test error in-sample training optimism

$$\text{Err}(\mathcal{T}) \approx \text{Err}_{\text{in}}(\mathcal{T}) = \overline{\text{err}}(\mathcal{T}) + \text{op}(\mathcal{T}) \approx \overline{\text{err}}(\mathcal{T}) + E_{\mathbf{y}}[\text{op}] = \overline{\text{err}}(\mathcal{T}) + \frac{2d\sigma^2}{N}$$

where $d = \text{tr}(\boldsymbol{H})$ denotes the effective degrees of freedom.
Remember that $\hat{y} = \boldsymbol{H}y$. Then the effective number of parameters is defined as $\text{tr}(\boldsymbol{H})$ and is also known as the effective degrees of freedom.

This helps us to distinguish between regression and residual effective degrees of freedom.

- $d = \text{tr}(\boldsymbol{H})$ is the regression effective degrees of freedom.
- The residual effective degrees of freedom are defined by $\text{tr}(\boldsymbol{I} - \boldsymbol{H})$.

H and I - H are basis for orthogonal spaces.

4.2 Cross Validation

Introduction

$Err(\tau) \approx \overline{err}(\tau) + \frac{2d\sigma^2}{N}$, $\frac{2d\sigma^2}{N} \geq 0$

- The test error can be easily calculated if a designated test set is available.
- On the contrary, the training error can be easily calculated on the training dataset; however, this can dramatically underestimate the test error.
- Cross-Validation is a popular method for estimating the average test error Err.
- Later we will see methods that make a mathematical adjustment to the training error rate, to estimate the test error rate. Here, we analyse methods based on resampling.

The validation set approach

Note: Based on "The Elements of Statistical Learning", if we are in a data-rich situation, the best approach for both problems is to randomly divide the dataset into three parts: a **training set**, a **validation set**, and a **test set**.

- The **training set** is used to fit the models;
- The **validation set** is used to **estimate prediction error** for model selection. This helps in choosing the best model among different models or configurations.
- The **test set** is used for **assessment of the generalization error** of the final chosen model.

However, since data are often scarce, this is usually not possible.

Therefore, suppose we take the dataset at hand and randomly divide it into two parts

- Training set
- Validation set

We do as follows:

- The model is fit on the training set, and then used to predict the validation set.
- The resulting validation set error rate provides an estimate of the test error rate.

Example: Cheddar dataset (R Code)

Consider the `cheddar` dataset from the `faraway` package in `R`.

- We fit multiple linear models including one or more of the following covariates `ace`, `h2s` and `lac`
- calculate the corresponding MSEs by randomly dividing the dataset into two parts.

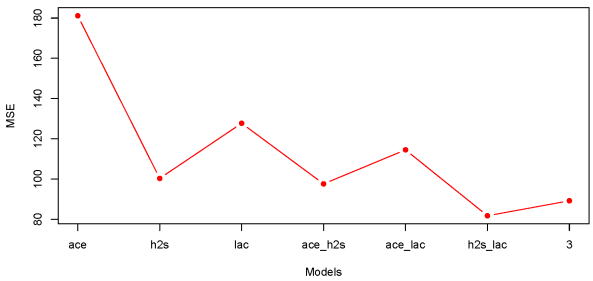


Figure 3: MSE values obtained for the `cheddar` dataset in order to estimate the test errors.

- It is possible to obtain different values of MSE.
- The validation set approach is easy both to implement and conceptually, but it has two main drawbacks:
  - The validation estimate of the test error rate can be very variable
  - Only a part of the observations is used to fit the model, i.e. the sample size is much smaller than the actual sample size

Leave-one-out cross-validation

To estimate the minimum test MSE using the training data use Cross-Validation (CV).

- CV directly estimates out of sample predictive performance.
- For many ML methods is the only method for the performance comparison.
- CV is very general and can be used with any type of predictive modelling.

Leave-One-Out Cross-Validation (LOO-CV)

LOO-CV is closely related to the validation set approach and involved splitting the dataset into two parts.

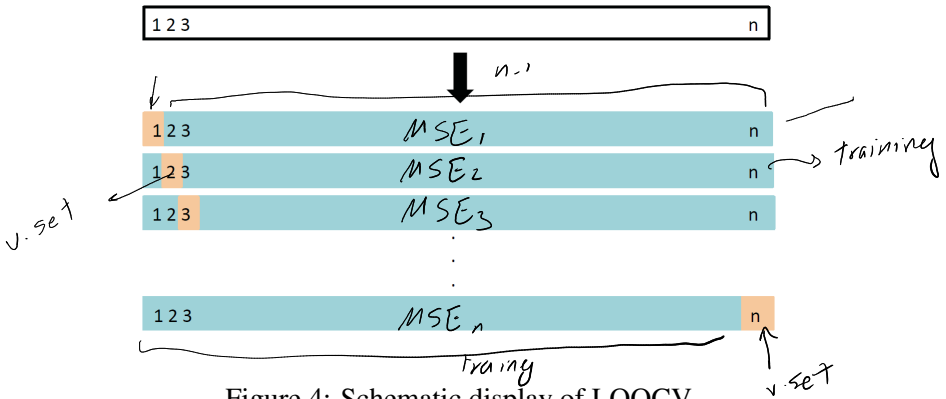


Figure 4: Schematic display of LOOCV

$$\frac{1}{n}\sum MSE_i = CV$$ LOO-CV estimate of the test MSE

Cont. Leave-one-out cross-validation

Step 1:

- Split the dataset to $\{(x_1, y_1)\}$ and $\{(x_2, y_2), \ldots, (x_N, y_N)\}$
- The statistical method is fit on the $N - 1$ observations of the training set
- The prediction $\hat{y}_1$ is compared on the single testing observation y_1
- Compute $\text{MSE}_1 = (y_1 - \hat{y}_1)^2$

Step 2:

- Split the dataset to $\{(x_2, y_2)\}$ and $\{(x_1, y_1), (x_3, y_3), \ldots, (x_N, y_N)\}$
- The statistical method is fit on the $N - 1$ observations of the training set
- The prediction $\hat{y}_2$ is compared on the single testing observation y_2
- Compute $\text{MSE}_2 = (y_2 - \hat{y}_2)^2$

⋮

Eventually, a vector $\text{MSE}_1, \text{MSE}_2, \ldots, \text{MSE}_N$ is produced and **the LOO-CV estimate of the test MSE** is

$$\text{CV}_{(N)} = \frac{1}{N} \sum_{i=1}^{N} \text{MSE}_i \tag{4.2.1}$$

Cont. Leave-one-out cross-validation

Advantages of LOO-CV

- It has less bias than the validation set approach because at each iteration $N - 1$ observations are used to fit the model
- The randomness due to how the dataset is split vanishes

Drawback of LOO-CV

- it is potentially expensive to implement since the model has to be fitted N times.

Example: Cheddar dataset (R Code)

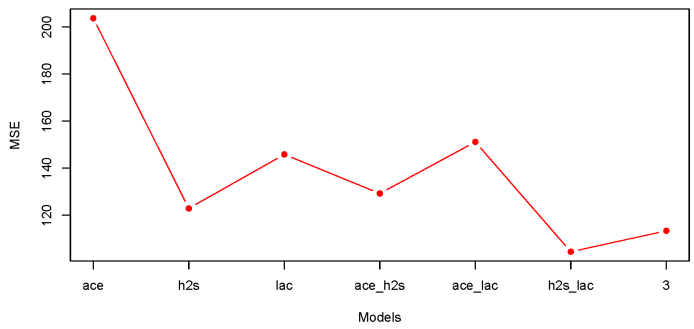


Figure 5: LOOCV error curves obtained for the cheddar dataset in order to estimate the test errors.

Cont. Leave-one-out cross-validation

We know that in **least squares regression**, $\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ and the predicted values are

$$\hat{\mathbf{y}} = \mathbf{X}^\top \hat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{y}$$

where $\mathbf{H}$ is the hat (projection) matrix, therefore, the CV score can be written

$$\mathrm{CV}_{(N)} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2 \qquad (4.2.2)$$

where h_i is the leverage, i.e. the i-th element on the diagonal of $\mathbf{H}$.

- This is the same as the ordinary MSE, except that each residual is divided by $1 - h_i$.
- The leverage represents the amount that an observation influences its own fit.

m-fold cross-validation

In m-fold CV, the data is randomly split into m roughly equal parts.

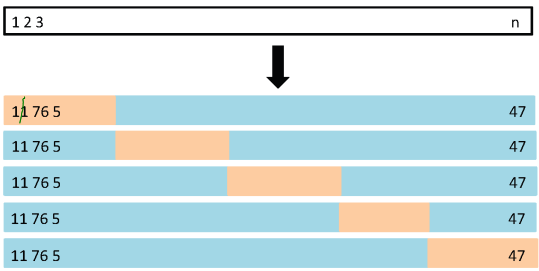


Figure 6: Schematic display of a 5-fold cross validation.

Consider the m equal sub-samples:

$$\{y_1^1, \ldots y_{n_1}^1\} \qquad \ldots \qquad \{y_1^m, \ldots, y_{n_m}^m\}$$

where $n_1, \ldots, n_m$ are approximate of the same size.

- x_j^i: the vector of predictor values corresponding to the response y_j^i,
- $y_{-i} = \{y_j^k : k \neq i, j = 1, ..., n_k\}$: the set of responses obtained by excluding the ith of the m sub-samples from the full set of responses,
- $\hat{y}_{-i}(x_j^k)$ be the prediction at x_j^k obtained by fitting to the responses y_{-i}.

Cont. m-fold cross-validation

For m-fold cross-validation, compute

$$\mathrm{CV}_{(m)} := \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}(y^i_j - \hat{y}_{-i}(x^i_j))^2.$$

A 5-step summary of the m-fold Cross-Validation is

1. Leave out the ith subsample
2. Fit the model on the remaining dataset
3. Predict the left-out observations
4. Sum up the squared prediction errors
5. Repeat this for every subsample
6. At the end, average the prediction errors.

Note: The LOO-CV is a special case of m-fold CV, where $m = N$.

Cont. m-fold cross-validation

Advantages:

- Computational: LOO-CV implies fitting the model N times, while m-fold CV implies fitting the model only m times
- Advantages for the bias-variance trade-off
  - LOO-CV gives **almost unbiased estimates** of the test error (model fitted with $N-1$ observations), while m-fold CV introduces **more bias** (model fitted with $\frac{(m-1)N}{m}$ observations).
  - When performing LOO-CV, the model is fitted N times (**highly correlated sets**). For m-fold CV, the output of m fitted models are averaged (**less correlated sets**). The output of highly correlated variables have higher variance than the mean of many variables less correlated.
  - m-fold CV tends to produce outputs, less variable than LOO-CV.

Example: Cheddar dataset (R Code)

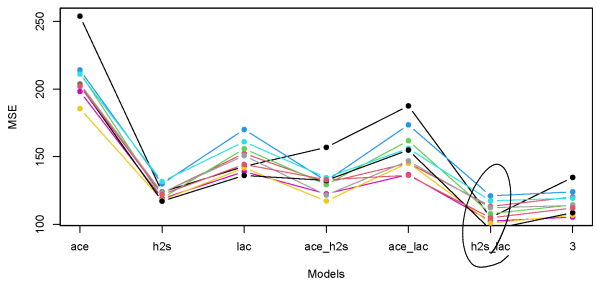


Figure 7: 10 repetitions of a 5-fold CV error curves obtained for the `cheddar` dataset in order to estimate the test errors.

4.3 Information Criteria

Introduction

When we compare different models, we would like to select the best, i.e. the one leading to the lowest test error. RSS and R^2 are not suitable for this task, because they are based on the training sample and the training error can be a poor estimate of the test error.

In order to select the best model we can

- Estimate the test error directly through cross-validation
- Estimate the test error by adjustment of the training error

Mallow's C_p

How to adjust the training error for the model size?

For linear models with p predictors estimated using least squares, Mallow's C_p is

$$C_p = \overline{\text{err}} + \frac{2p\hat{\sigma}^2}{N},$$

- $\overline{\text{err}} = \text{RSS}/N$
- $\hat{\sigma}^2$: estimate of the variance of the noise ε in the model $Y = f(X) + \varepsilon$.

C_p :

- gives an estimate of Err_{in} (In-sample prediction error)
- a criterion for model selection: Choose the model complexity with minimal C_p.
- adds a penalisation $\frac{2p\hat{\sigma}^2}{N}$ to the training error, because it tends to underestimate the test error
- the penalisation increases as the number of predictors increases.

AIC and BIC

Consider a range of information criteria, since

- sometimes you will obtain conflicting information based on some of these criteria.
- the more information criteria you use in your analysis, the more insight you will possess for model selection.

Akaike Information Criterion

- For a linear model, RSS equals (up to a constant) the negative log-likelihood.
- For likelihood based models, Mallow's Cp generalises to the Akaike Information Criterion (AIC):

$$\text{AIC} = -2\sum_{i=1}^{N} \ell(\hat{\boldsymbol{\beta}}, y_i) + 2d, \tag{4.3.1}$$

- it aims to minimise the Generalisation Error, asymptotically for large N.
  - d is the number of estimated parameters in the fitted model (for Gaussian regression, $d = p + 2$ $(\beta_0, \beta_1, \ldots, \beta_p, \sigma^2)$.

Hurvich and Tsai (1989) developed AICc, a version of the AIC:

$$\text{AICc} = \text{AIC} + \frac{2d(d+1)}{N-d-1}$$

- is asymptotically equivalent to AIC as $N \to \infty$
- corrects for a bias at small sample sizes:
- is recommend that AICc be used instead of AIC unless $N/d > 40$.

Bayesian Information Criterion

The Bayesian information criterion (BIC) is

$$\text{BIC} = -2\sum_{i=1}^{N} \ell(\hat{\beta}, y_i) + d \log N. \tag{4.3.2}$$

- It penalises complexity more strongly than AIC if $N \geq 8$, since then $\log N > 2$.
- The model with the lowest BIC corresponds to the model with the highest log-posterior probability, assuming an indifferent prior.

R^2-adjusted criterion

R^2 is unsuitable for the choice of model parameters, since it typically increases with every added parameter. The adjusted R^2 is

$$R^2_{adj} = 1 - (1 - R^2)(n - 1)/(n - p - 1)$$

and using the fact that $R^2 = 1 - RSS/TSS$

$$R^2_{adj} = 1 - \frac{RSS/(n - p - 1)}{TSS/(n - 1)}$$

where p is the number of predictors in the current model. It can be used for model selection, but still tends to "overfit".

Example: Cheddar cheese (R Code)

4.4 Variable Selection

Variable Selection

The task of determining which predictors are associated with the response, in order to fit a single model involving only those predictors, is referred to as **variable selection**.

There are two distinctly different approaches to choosing the potential subsets of predictor variables,

- **Best Subset Selection**
- **Stepwise Methods**

Best Subset Selection

This approach considers all 2^m possible regression equations and identifies the subset of the predictors of a given size that maximises a measure of fit OR minimises an information criterion.

- With a **fixed** number of terms in the regression model, R^2_{adj}, AIC, AICc and BIC agree that the best choice is the set of predictors with the smallest value of the residual sum of squares.
- When the comparison is across models with different numbers of predictors, the four methods (R^2_{adj}, AIC, AICc and BIC) can give entirely different results.

Example: Best Subset Selection (R Code)

We want to predict a baseball player's `Salary` by various statistics associated with performance in the previous year. We will use the *regsubsets()* function to perform best subset selection by identifying the best model that contains a given number of predictors, where best is quantified using RSS.

Stepwise Model Selection

This approach is based on examining just a sequential subset of the 2^m possible regression models.

- **Backward elimination** starts with all potential predictors in the regression model. Then, at each step, it deletes the predictor variable such that the resulting model has the lowest value of an information criterion (removing the predictor with the largest p-value).
- **Forward selection** starts with no potential predictors in the regression equation. Then, at each step, it adds the predictor such that the resulting model has the lowest value of an information criterion (adding the predictor with the smallest p-value).

Backward elimination and forward selection:

- do not necessarily find the model that minimises the information criteria across all 2^m possible predictor subsets,
- there is no guarantee that they will produce the same final model.

Example: Backward and forward selection (R Code)

- Cheddar dataset
- Sydney maximum temperature

Ridge Regression

Introduction

Where there are many **correlated variables** in a linear regression model,

- their coefficients can become poorly determined and exhibit **high variance**.
- A wildly large positive coefficient on one variable can be cancelled by a similarly large negative coefficient on its correlated cousin.
- By imposing a **size constraint** on the coefficient, this phenomenon is prevented from occurring.

Model selection effectively sets some β_j equal to zero. Based on this idea, **ridge regression imposes a size constraint** on the coefficients, that is, "**shrinks**" them towards zero. Benefits are similar to model selection:

- **Model complexity** is effectively **reduced**, leading to a **smaller variance**,
- if done properly, leads to a **smaller mean squared prediction error**.

shrink β_j to zero $\neq$ zero

Ridge Regression

The task is to minimize

$$\mathrm{RSS}(\lambda) = \|\boldsymbol{y} - a_0\mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 = \sum_{i=1}^{n}(y_i - a_0 - x_i^\top\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \qquad (4.5.1)$$

with respect to a_0 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$.

- $x_i = (x_{i1}, \ldots, x_{ip})$, and $\boldsymbol{X}$ does not have a leading column of 1's.
- The parameter $\lambda > 0$ controls the degree of shrinkage.
- The **first term** is **the least squares criterion**.
- The **additional term** $\lambda\sum_{j=1}^{p}\beta_j^2$ **penalizes large coefficient values**
  - There is no penalty for the intercept term.

How much large coefficient values are penalised is controlled by **the shrinkage parameter** λ:

- $\lambda = 0$: the ridge estimator is just the ordinary least squares estimator
- $\lambda = \infty$: the fitted model is a constant model.

Cont. Ridge Regression

An equivalent way of writing the ridge estimate of $\boldsymbol{\beta}$ is to minimise the RSS subject to

$$\sum_{j=1}^{p} \beta_j^2 \leq s. \tag{4.5.2}$$

This makes the size constraint on the parameters explicit. There is a one-to-one correspondence between the shrinkage parameter λ and s.

Example: Polynomial Fitting

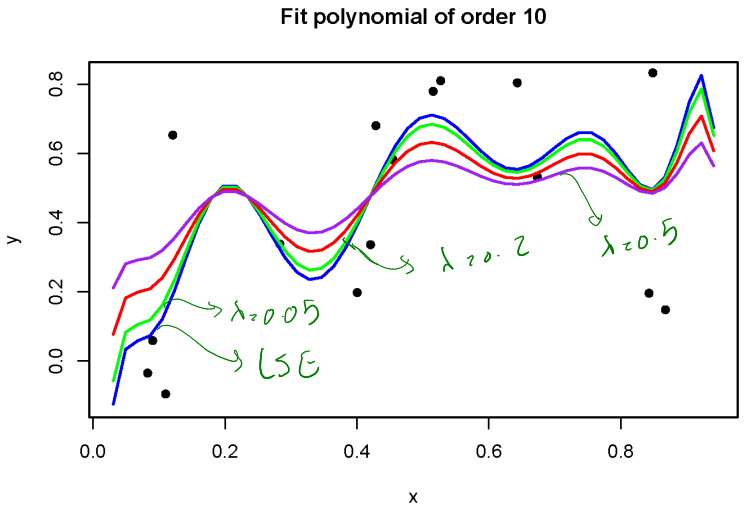
Recall the effects of fitting an overly complex polynomial. We considered a simulated data set of size $n = 20$ from the model

$$y_i = x_i + \epsilon_i$$

where $\epsilon_i \sim N(0, 0.25^2)$. A polynomial model of order k is

$$y_i = a_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_k x_i^k + \epsilon_i.$$

The true order of the polynomial here is 1. For our simulated data, consider a tenth-degree polynomial fit by ordinary least squares as well as ridge estimators for $\lambda = 0.05, 0.2, 0.5$.



λ small
↓
LSE

λ larg
↓
minimal model

Figure 8: Fitted polynomials using least squares (blue) and ridge regression with $\lambda = 0.05$ (green), 0.2 (red) and 0.5 (purple).

Ridge Regression: Centring and scaling

Let $\bar{x}_j = \sum_{i=1}^{n} x_{ij}/n$, the scaled predictors are defined as

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

The two models

$$y_i = a_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i \quad \text{and} \quad y_i = a_0 + \beta_1 z_{i1} + \ldots + \beta_p z_{ip} + \epsilon_i$$

are **equivalent**, in the following sense:

- Shifts of $\boldsymbol{X}_j$ (by $\bar{x}_j$) result in a shift in the intercept β_0, but leave the other coefficients unchanged;
- Division of $\boldsymbol{X}_j$ by S_j effectively multiplies β_j by S_j.
- Least squares estimates in one scale linearly rescale to least squares estimates in another.

This means that the choice of scale is irrelevant for inference and prediction purposes in linear models without shrinkage.

Cont. Ridge Regression: Centring and scaling

- The ridge solutions are NOT equivalent under scaling of the inputs, and so one normally standardises the inputs before solving (4.5.1).
- It can be shown that the solution to (4.5.1) can be separated into two parts, after reparametrization using centered inputs: each x_{ij} gets replaced by $z_{ij} = x_{ij} - \bar{x}_j$.
  - We estimate a_0 by $\bar{y}$.
  - The remaining coefficients get estimated by a ridge regression without an intercept, using the centred z_{ij}.
- Henceforth we assume that this centering has been done, so that the input matrix Z has p (rather than $p+1$) columns.
- The ridge regression criterion is than

$$\mathrm{RSS}(\lambda) = \|\boldsymbol{y}^{(c)} - \boldsymbol{Z}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

with solution

$$\hat{\beta} = (\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{I})^{-1} \boldsymbol{Z}^\top \boldsymbol{y}^{(c)},$$

where $\boldsymbol{I}$ is the $p \times p$ identity matrix and $\boldsymbol{y}^{(c)} = \boldsymbol{y} - \bar{\boldsymbol{y}}$.

  - Note that the solution adds a positive constant to the diagonal of $\boldsymbol{Z}^\top \boldsymbol{Z}$ before inversion. This makes the problem nonsingular, even if $\boldsymbol{Z}^\top \boldsymbol{Z}$ is not of full rank.
  - This means that the least squares estimate exists if the columns of $\boldsymbol{Z}$ (and equivalently $\boldsymbol{X}$) are not linearly independent, or even if $p > N$. In fact, this was the primary motivation for ridge regression when it was first introduced in Hoerl & Kennard (1970).

Example: Hospital manpower Data

Variance reduction through shrinkage

Singular value decomposition (SVD) of the centred input matrix $\boldsymbol{Z}$ gives us some additional insight into the nature of ridge regression. We can compute SVD of the $N \times p$ centered data matrix $\boldsymbol{Z}$ as

$$\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top},$$

where

- $\boldsymbol{U}$: a $N \times p$ matrix with orthonormal columns that span the column space of $\boldsymbol{Z}$,
- $\boldsymbol{V}$ is a $p \times p$ orthogonal matrix,
- $\boldsymbol{D} = diag(d_1, d_2, \cdots, d_p)$ such that $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$.

$U^T U = I$

$V^T V = diag(\ldots)$

Cont. Variance reduction through shrinkage

Using the SVD we can rewrite the expression for $\hat{\boldsymbol{\beta}}$ as follows

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{I})^{-1} \boldsymbol{Z}^\top \boldsymbol{y} \\ &= (\boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^\top + \lambda \boldsymbol{V}\boldsymbol{V}^\top)^{-1} \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{y} \\ &= (\boldsymbol{V}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})\boldsymbol{V}^\top)^{-1} \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{y} \\ &= \boldsymbol{V}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{y}.\end{aligned}$$

Consequently,

$$\begin{aligned}\hat{\boldsymbol{y}} = \boldsymbol{Z}\hat{\boldsymbol{\beta}} &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \boldsymbol{V}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{y} \\ &= \boldsymbol{U}\boldsymbol{D}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}\boldsymbol{U}^\top \boldsymbol{y}.\end{aligned}$$

$\boldsymbol{U}^\top \boldsymbol{y}$: coordinates of vector $\boldsymbol{y}$ in the space spanned by $\boldsymbol{U}$

$\boldsymbol{D}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D} = \operatorname{diag}\left(\frac{d_j^2}{d_j^2 + \lambda}\right)$

We note that

- $\boldsymbol{D}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}$ is a diagonal matrix with elements given by $\frac{d_j^2}{d_j^2 + \lambda}$.
- the vector $\boldsymbol{U}^\top \boldsymbol{y}$ is the coordinates of the vector $\boldsymbol{y}$ in the basis spanned by the p-columns of $\boldsymbol{U}$.

Thus

$$\hat{\boldsymbol{y}} = \boldsymbol{Z}\hat{\boldsymbol{\beta}} = \sum_{j=1}^{p} \boldsymbol{U}_j \left(\frac{d_j^2}{d_j^2 + \lambda}\right) \boldsymbol{U}_j^\top \boldsymbol{y}$$

and the inner products $\boldsymbol{U}_j^\top \boldsymbol{y}$ are scaled by the factors $\frac{d_j^2}{d_j^2 + \lambda}$ in the ridge regression.

Cont. Variance reduction through shrinkage

- The hat matrix $\boldsymbol{H}_\lambda$, defined via $\boldsymbol{H}_\lambda \boldsymbol{y} = \hat{\boldsymbol{y}}$ depends on λ and is equal to

$$\boldsymbol{H}_\lambda = \boldsymbol{Z}(\boldsymbol{Z}^\top \boldsymbol{Z} + \lambda \boldsymbol{I})^{-1} \boldsymbol{Z}^\top = \boldsymbol{U}\boldsymbol{D}(\boldsymbol{D}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}\boldsymbol{U}^\top.$$

- The fitted values for ridge regression satisfy

$$\mathrm{Cov}(\hat{\boldsymbol{y}}) = \sigma^2 \boldsymbol{H}_\lambda^2 = \sigma^2 \boldsymbol{U}\mathrm{diag}\left(d_1^4/(d_1^2+\lambda)^2, \ldots, d_p^4/(d_p^2+\lambda)^2\right)\boldsymbol{U}^\top.$$

- The shrinkage indeed reduces the predictive variance.
- The effective number of parameters for ridge regression is

$$\mathrm{tr}(\boldsymbol{H}_\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}.$$

- as λ increases from 0 to $+\infty$, the effective number of parameters decreases continuously and monotonically from p to 0.

The Lasso

- The lasso is a shrinkage method like ridge regression, with subtle but significant differences.
- The lasso estimate is defined by the following modification of the ridge regression optimisation problem:

$$\text{RSS}(\lambda) = \|\boldsymbol{y} - a_0\mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{n}(y_i - a_0 - x_i^\top\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

- An equivalent formulation of the problem is to minimise the RSS subject to

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

- The same considerations on the scaling of predictors as for ridge regression apply to the lasso model.
- The difference between ridge and lasso lies entirely in the form of the penalty term, which uses the L^1-norm instead of the (Euclidean) L^2-norm; see Figure 9.

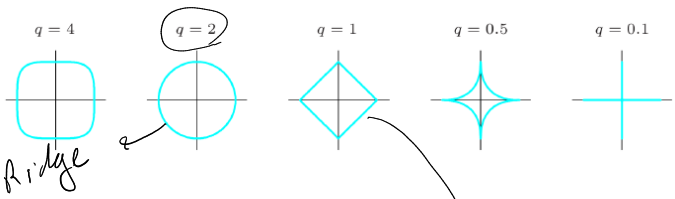


Figure 9: The contours $\|x\|_q = 1$, for different values of q. Note that $q = 1$ is the only value for which the L^q-norm is both convex and shows cusps.

Cont. The Lasso

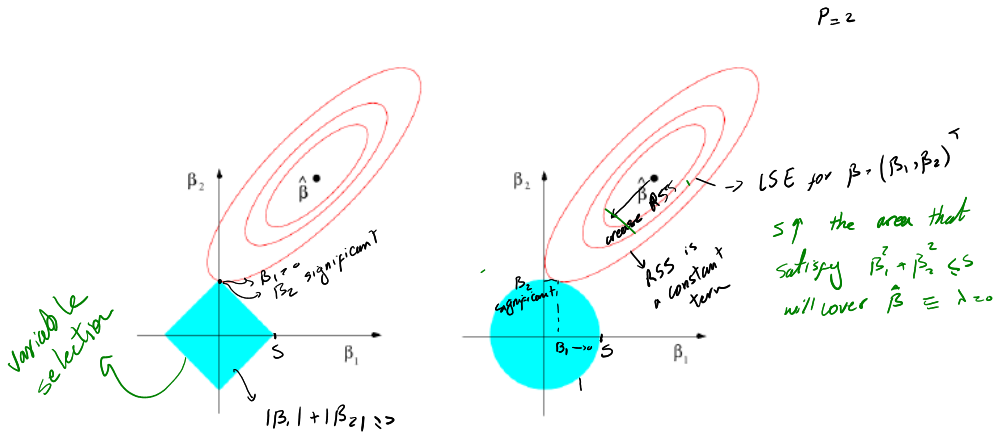


Figure 10: Restricted parameter set (blue) and contours of the least squares error function (red). The lasso (left) shrinks β_1 to 0 (exactly), whereas ridge regression (right) does not.

Comments:

- $\hat{\beta}$ is the least squares estimate.
- If s is sufficiently large, the constraint regions would contain the LS estimates (corresponding to the case $\lambda = 0$)
- The red ellipses represent the regions of constant RSS: going far from $\hat{\beta}$ would increase RSS.
- Lasso and ridge estimates are given by the first point at which an ellipse touch the constraint region.
- Since lasso, differently from ridge, has corners on the axes, the ellipses are more likely to touch the constraint region on an axis.

As illustrated for $p = 2$ parameters in Figure 10, the lasso formulation of the restricted parameter space favours values for $\boldsymbol{\beta}$ which lie at a cusp, meaning that one or more coordinates of $\boldsymbol{\beta}$ are set to 0. A similar statement holds in higher dimensions. The lasso hence both selects and shrinks the coefficients $\boldsymbol{\beta}$.

Example: Lasso Fit in R (Hospital manpower data)

Example: Lasso vs. Ridge Regression (Hitters dataset)

4.7 Model Selection