Introduction to Statistical Learning
with applications in Python

Based on "Introduction to Statistical Learning, with applications in R" by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Resampling Methods

Cross-Validation for Regression, Validation Set, Leave-One-Out, k-Fold, Cross-Validation for Classification, Bootstrap

Kurt Rinnert

Physics Without Frontiers

Abstract

We already mentioned the importance of evaluating models on test data sets that were not used in the training phase. We now expand on this idea.
Resampling methods are based on the idea of repeatedly drawing samples from a training data set and refitting the model in order to better evaluate the fitted model.
These methods used to be computationally prohibitive because they involve multiple model fits. This is no longer a problem due to cheaply available computing resources.

Overview

- Cross-Validation
- The Validation Set Approach
- Leave-One-Out Cross-Validation (LOOCV)
- k-Fold Cross-Validation
- Cross-Validation for Classification
- The Bootstrap Method

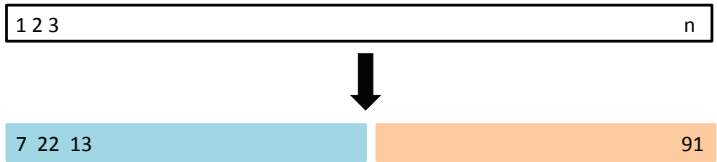
All these methods provide *estimates* of the test error.

Cross-Validation

- We have already discussed the difference between *training error rate* and the *test error rate.*
- We can easily calculate the test error if we have a sufficiently large test data set.
- In the absence of a dedicated test data set we can use a subset of the training data to estimate the test error.
- This subset is called the *validation set* or *hold-out set*.

We first assume regression and deal with classification later.

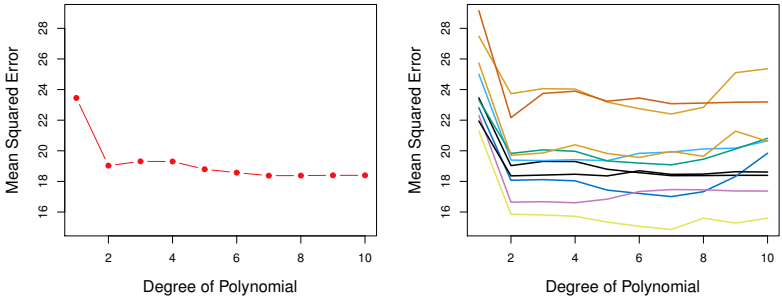
The Validation Set Approach



- We *randomly* split the training data into a training set and a validation set.
- The model is then fit on the training set and evaluated on the validation set.
- In case of a quantitative response, we typically use the MSE on the validation set to evaluate the model.

The distinction between *validation set* and *test set* is rather subtle.

The Validation Set Approach



- The *estimate* of the test error derived from the validation set can be highly variable.
- The validation error tends to *overestimate* the test error rate due to the lower number of observations.

These issues are addressed by the various *cross-validation* methods.

Leave-One-Out Cross-Validation

The LOOCV repeatedly uses *one* observation for validation.

Leave-One-Out Cross-Validation

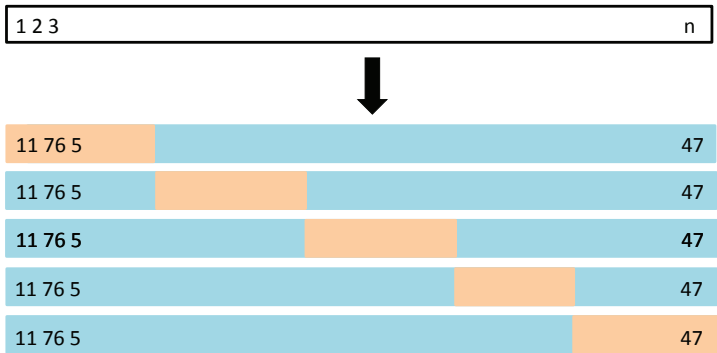
- The model fit is performed n times.
- Where, as usual, n is the number of training observations.
- We then average the MSE's from the n fits:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i$$

- That is, we sample the test set population and compute and estimate the test error.

The LOOCV has low bias but high variance.

k-Fold Cross-Validation



This approach repeatedly uses subsets of size n/k for validation.

k-Fold Cross-Validation

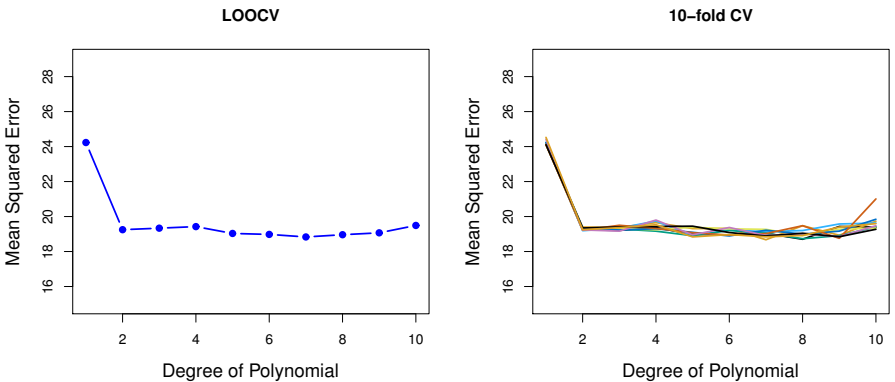
- Note that the training set is *randomly shuffled* before folding.
- The model fit is performed k times.
- We then average the MSE's from the k fits:

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i$$

- Again, we sample the test set population and compute and estimate the test error.

The bias-variance trade-off depends on the choice of *k*.

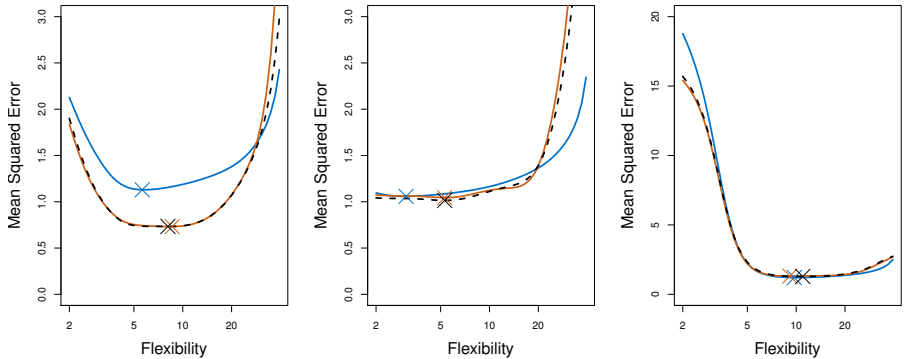
Leave-One-Out versus k-Fold Cross-Validation



LOOCV (left) and 10-fold cross validation (right) on `Auto`.

Values between 5 and 10 are typically good choices for for k.

Leave-One-Out versus *k*-Fold Cross-Validation



LOOCV and 10-fold cross validation on the simulated scenarios from lecture 2.

Close to the MSE minimum both methods have very similar results.

Cross-Validation for Classification

- Cross-Validation also works in the classification setting.
- The misclassification rate takes the role of the MSE.
- For LOOCV the cross-validation error rate is

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_1)$$

with

$$I(y_i \neq \hat{y}_i) = \begin{cases} 1 & \text{if } y \neq \hat{y}_i \\ 0 & \text{if } y = \hat{y}_i \end{cases}$$

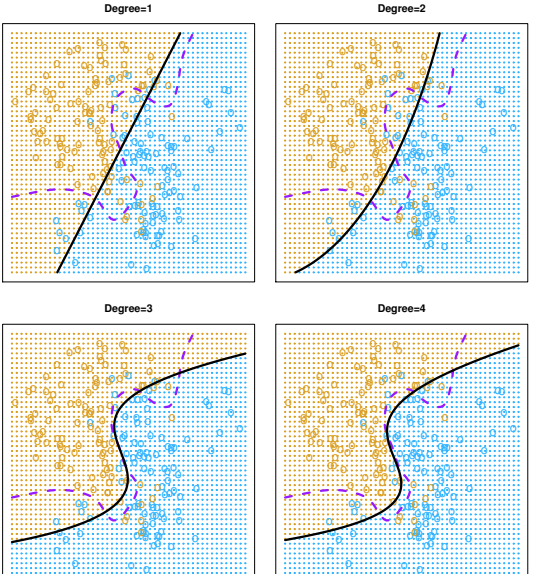
We can evaluate classification models just like regression models.

Cross-Validation for Classification

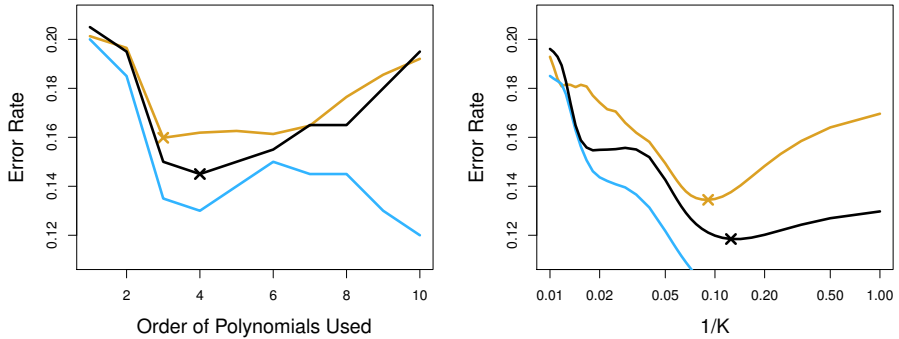
- The figures show four different polynomial logistic regression models.
- Non-linear terms can be added to the logistic model just like in linear regression.
- For example, the quadratic (Degree=2) model is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$$

A degree > 3 doesn't help much anymore.



Cross-Validation for Classification



- The figures show various polynomial models (left) and an KNN classifier (right).
- The test error (orange), training error (blue) and cross-validation error (black) are shown.
- The *K* in the right plot denotes the neighbours in KNN.

The test error is obtained from a simulated test sample.

The Bootstrap Method

- *Bootstrap* is widely applicable method to assess model uncertainties.
- We demonstrate it on a simulated toy example.
- Suppose we have some money we want to invest.
- We would like to invest a fraction α in an asset with return X and the rest, $1 - \alpha$, in an asset with return Y such that the risk is minimised.
- The risk is associated with the variance of the total return:

$$\mathrm{Var}\left(\alpha X + (1-\alpha)Y\right)$$

which is minimal when

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

In practice, we have to estimate σ_X^2, σ_Y^2 and σ_{XY} from the training data.

The Bootstrap Method

- In the toy simulation the true values are $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.25$ and $\sigma_{XY} = 0.5$.
- The α with minimal risk is then $\alpha = 0.6$.
- We simulate 1000 independent samples with 100 observations each to obtain

$$\hat{\alpha} = \overline{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996$$

and the error estimate

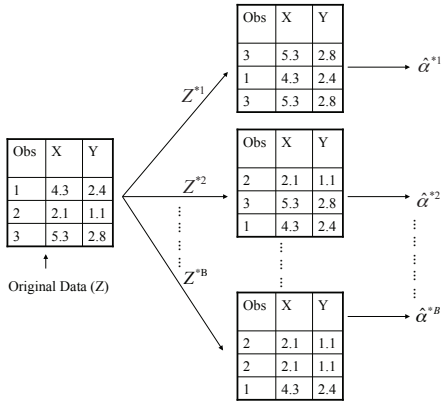
$$\widehat{\mathrm{SE}}(\overline{\alpha}) = \sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} \left(\hat{\alpha}_r - \overline{\alpha}\right)} = 0.083$$

In practice, we don't have a large number of independent samples.

The Bootstrap Method

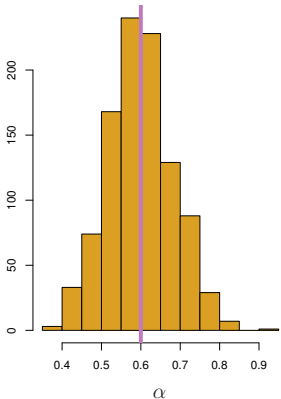
- In the absence of independent test samples we need a different solution.
- The *bootstrap* method creates test samples by repeatedly sampling from the training data.
- The sampling is done with *replacement*.
- That is, an observation can appear more than once in each sample.
- Sampling B times we obtain

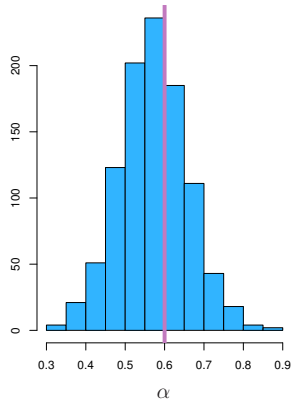
$$\widehat{\mathrm{SE}}(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}$$

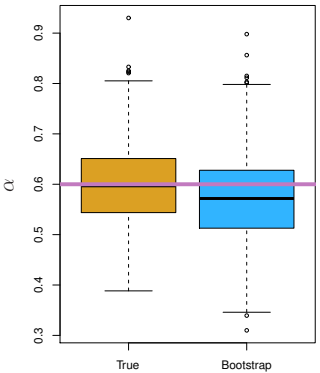


In practice, B needs to be sufficiently large, $\mathcal{O}(1000)$.

The Bootstrap Method







We can see that the bootstrap method does very well.