ECON3389 Machine Learning in Economics

Module 3: Cross Validation
Resampling Methods

Alberto Cappello

Department of Economics, Boston College

Fall 2024

Overview

Agenda:

- Cross-validation
- Bootstrap

Readings:

- ISLR Chapter 5

Resampling Methods

Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model

Cross Validation

- Model assessment: estimate the test error associated with a model in order to evaluate its performance
- Model selection: select the appropriate level of flexibility

Bootstrap

- Most commonly used to provide a measure of accuracy of a parameter estimate

Cross Validation

- As discussed in our first lecture, the more flexible is the model, the better it fits any given dataset, but the higher is the chance of overfitting — having a model be very good at fitting one particular dataset and very bad at fitting any other dataset.
- One solution is to use a *train* dataset for model's estimation and a *test* dataset for choosing the degree of model's flexibility.
- The basic idea is then to use the training data to estimate the model for various levels of flexibility, using only the training data:

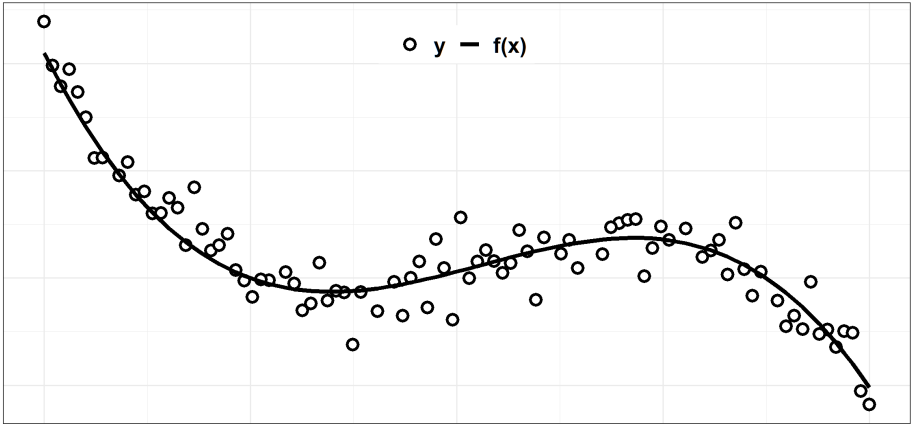
$$MSE_{Tr} = \frac{1}{n_{Tr}} \sum_{i \in \text{Tr}} \left[y_i - \hat{f}(x_i) \right]^2$$

- Then, in a second step, let the test data decide what the optimal model flexibility should be:

$$MSE_{Te} = \frac{1}{n_{Te}} \sum_{i \in \text{Te}} \left[y_i - \hat{f}(x_i) \right]^2$$

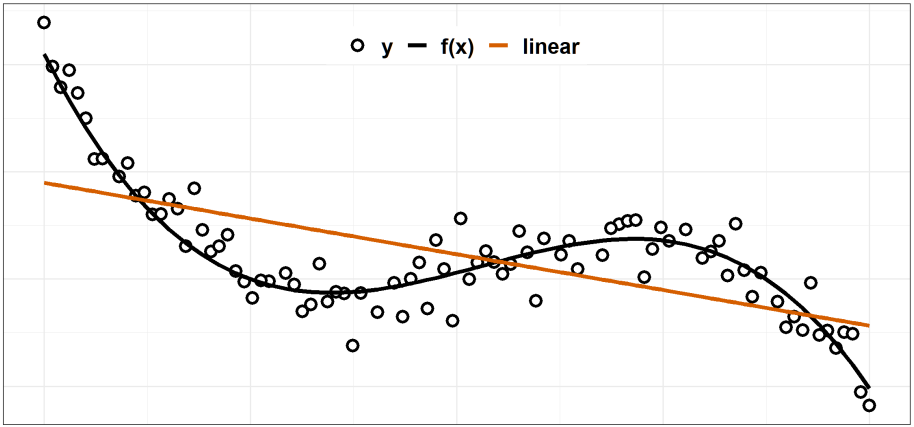
- The general rule is to pick the model that produces smallest *test error* (usually MSE), as that typically would mean the best bias-variance trade-off.

Overfitting



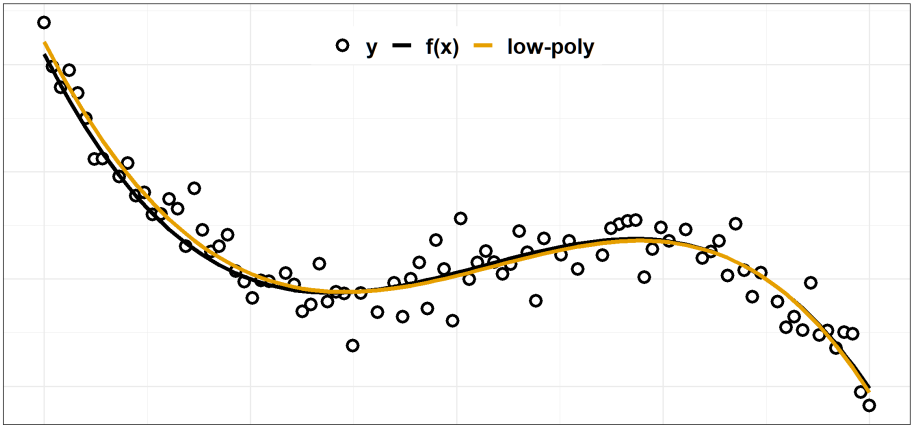
Black line is true $f(X)$, black circles are observed train data values of $y = f(x) + \epsilon$.

Overfitting



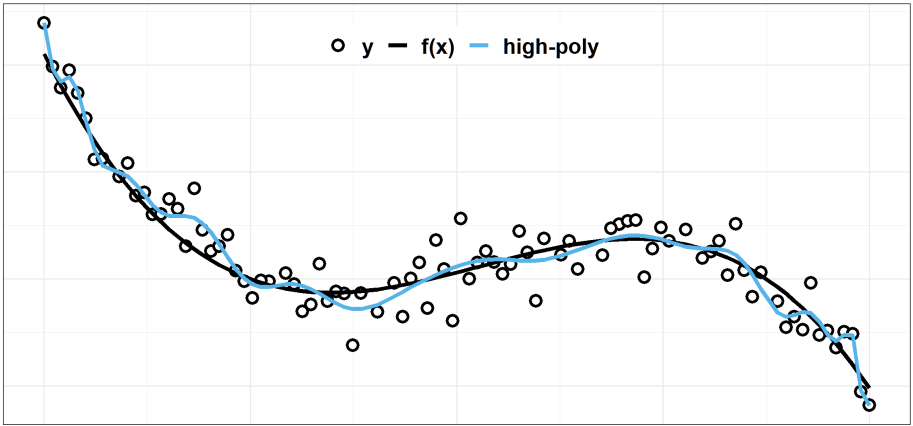
Even though true $f(X)$ is very smooth, a linear bit is clearly not flexible enough – it fails to follow both $f(X)$ and actual data points, but it does have a clear interpretation.

Overfitting



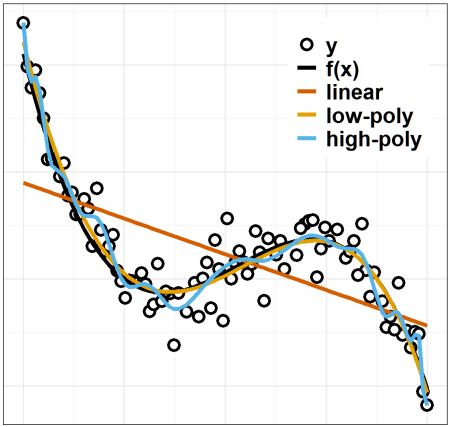
A more flexible low-order polynomial fit follows true $f(X)$ very closely, but not so much actual data points, yet providing somewhat clear interpretation.

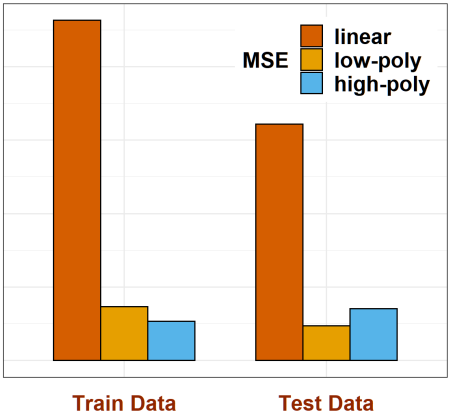
Overfitting



Very flexible high-order polynomial fit follows data points closely, but is too wiggly and thus deviates a lot from true $f(X)$. There is no clear interpretation for this fitted model.

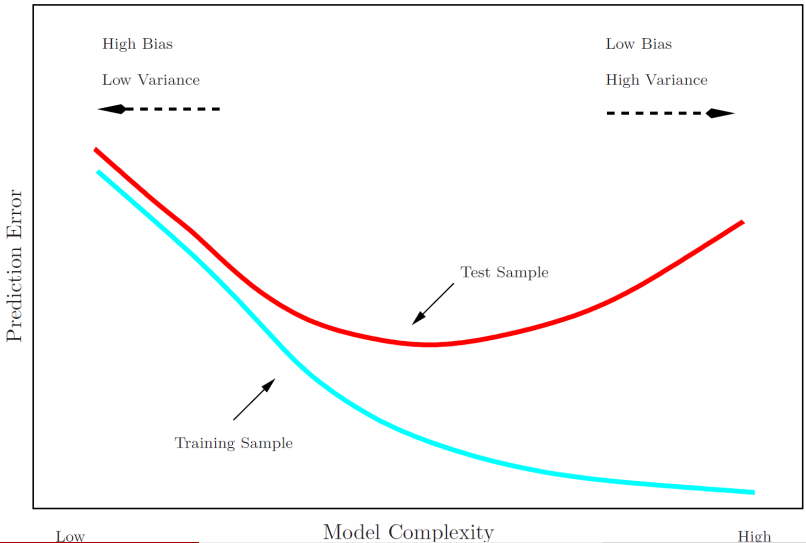
Overfitting





Small training MSE + Large test MSE $\rightarrow$ Overfitting

Train/Test Split



Bias-variance trade-off

- Suppose our test data Te consists of a single data point (x_0, y_0). Then

$$\mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \text{Var}\left(\hat{f}(x_0)\right) + \text{Bias}^2\left(\hat{f}(x_0)\right) + \text{Var}(\epsilon_0)$$

- The left hand side is the expected test MSE at x_0
- Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set
- Bias refers to the error that is introduced by approximating a real-life problem by a much simpler model
- Typically as the flexibility of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on MSE amounts to a bias-variance trade-off.

Train/Test Split

- Best solution: a large designated test set, ideally being sampled independently of train dataset. But such test data is rarely available.
- Some methods make a mathematical adjustment to the training error rate in order to estimate the test error rate without actually using test data. These include the *Cp statistic*, *AIC* and *BIC*. They will be discussed later in this course.
- We instead consider a class of methods that estimate the test error by holding out a subset of the training observations from the model fitting process, and then applying the statistical learning method of choice to those held out observations, using them as test data.

Validation-set approach

- Suppose we want to select the right model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + ... \quad (1)$$

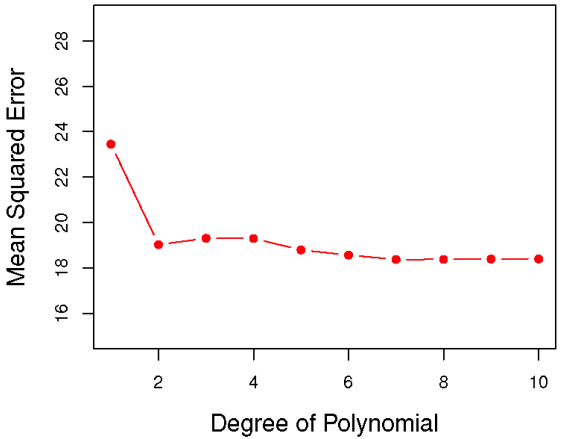
- We have already seen what is formally known as *validation set* approach — randomly dividing the available set of samples into a training set and a validation or hold-out set.
- The model is fit on the training set, and the fitted model is used to predict the responses for observations in the validation 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 in the case of a qualitative (discrete) response.

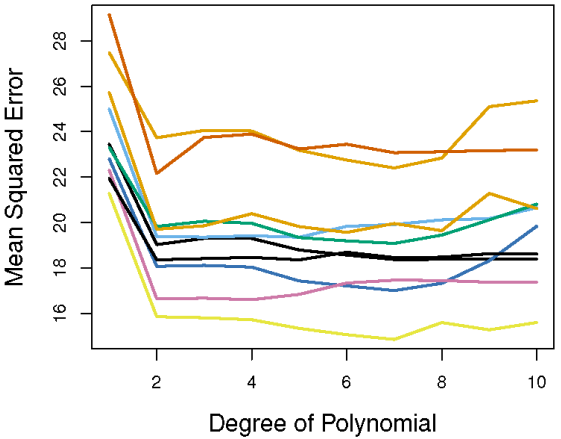
Validation-set approach: example

- Consider `Auto` dataset available in base R and a relationship between `mpg` and `horsepower`. A brief analysis shows this relationship is likely a non-linear one. But how much non-linearity is there? Would $\texttt{horsepower}^2$ be enough or should we add cubic or higher order terms as well?
- 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.
- We then fit each model on the training set and calculate its MSE on the validation set. And then we repeat this process 10 times, each time getting a different random split of our data.

Validation-set approach: example



Left panel: single validation split.



Right panel: multiple validation splits.

Validation-set approach: drawbacks

- 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.
- Because we are using less data, our fit may end up being less precise just because we use less information, therefore overestimating the test error for the model compared to the fit on the entire data set.

Cross-validation

- A far more widely used approach is to split (fold) the data randomly K times and validate our model across all K folds, hence the name *K-fold cross-validation*.
- Idea is to randomly divide the data into K equal-sized parts. We leave out part k that acts as our k^{th} validation set, fit the model to the total data in the remaining $K-1$ parts (combined), and then obtain predictions for the left-out k^{th} part.
- Repeat this process in turn for each part $k = 1, 2, \ldots, K$, and then aggregate the results (usually through averaging).

Cross-validation: details

- Let $C_1, C_2, \ldots, C_K$ be a set of K indices of the observations that describe which part of data goes into part k.
  - E.g. $C_1 = \{4, 5, 7\}, C_2 = \{1, 3, 6\}, \ldots$
- The aggregate model precision measure is then computed as

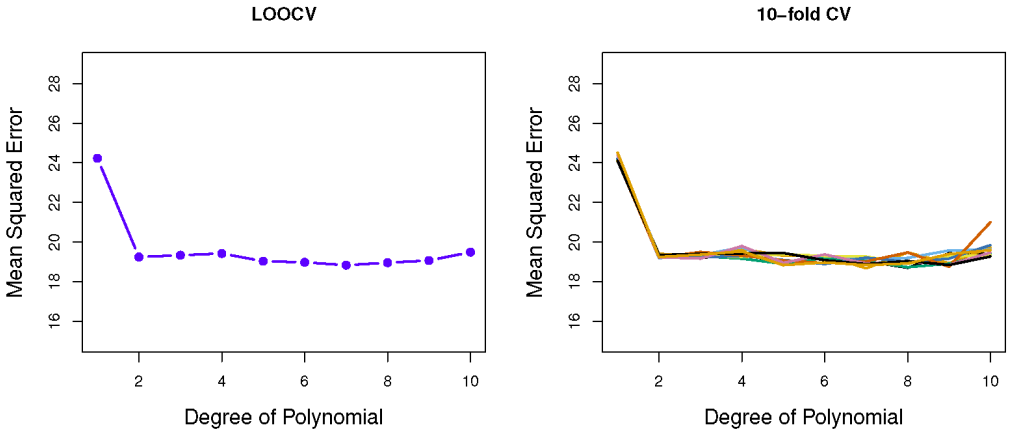
$$CV_{(K)} = \sum_{k=1}^{K} \frac{1}{K} \text{MSE}_k$$

where $\text{MSE}_k = \sum_{i \in C_k} \frac{(y_i - \widehat{y}_i)^2}{n_k}$ and $\widehat{y}_i$ is the fit for observation i, obtained from the model fitted on data with part k removed.

Leave-one-out cross-validation

- Setting $K = n$ yields n-fold or *leave-one-out cross-validation* (LOOCV).
- A major advantage of LOOCV over validation set approach is that it has far less bias. We repeatedly fit the model using training sets that contain $(n - 1)$ observations, almost as many as are in the entire data set
- However, it might be computationally expensive
- Overall LOOCV typically looses in bias-variance trade-off vs a K-fold CV, thus more common choice is $K = 5$ or $K = 10$.

Auto data revisited



Left panel shows a single LOOCV split, right panel shows multiple 10-fold CV splits.

Cross Validation vs Test MSE

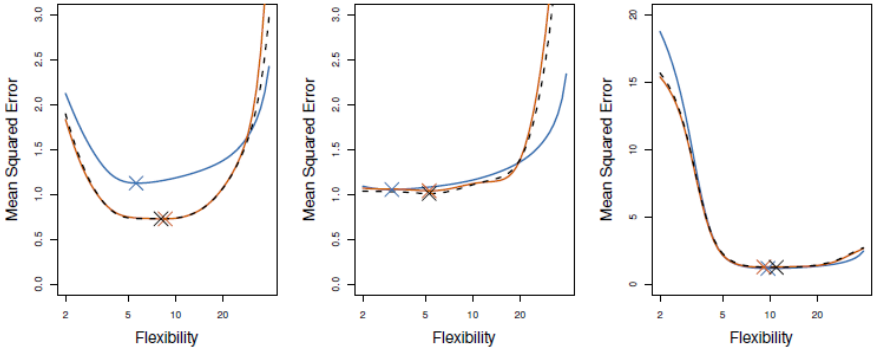


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (*left*), 2.10 (*center*), and 2.11 (*right*). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the* 10*-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

Bias vs Variance trade-off: LOOCV vs K-fold

- We are trying to estimate the *test MSE*
- K-fold cross validation has computational advantages over LOOCV
- A more important advantage of K-fold CV is that it often gives more accurate estimates of the *test error rate*
- LOOCV will give approximately unbiased estimates of the test error since each training set contains $(n-1)$ observations, almost as many as are in the entire data set
- K-fold CV will generate an intermediate level of bias
- However, LOOCV estimate of MSE has higher variance
- When we perform LOOCV, we are in effect averaging the outputs of n fitted models, each of which is trained on an almost identical set of observations; therefore, these outputs are highly (positively) correlated with each other
- In contrast, when we perform k-fold CV, we are averaging the outputs of k fitted models that are somewhat less correlated with each other, since the overlap between the training sets in each model is smaller

Cross-validation on classification problems

- Let $C_1, C_2, \ldots, C_K$ be a set of K indices of the observations that describe which part of data goes into part k.
  - E.g. $C_1 = \{4, 5, 7\}, C_2 = \{1, 3, 6\}, \ldots$
- The aggregate model precision measure is then computed using the misclassification rate

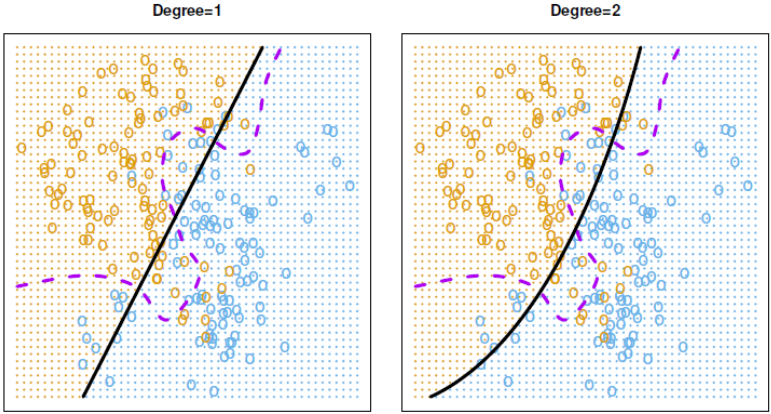
$$CV_{(K)} = \sum_{k=1}^{K} \frac{1}{K} \text{Err}_k$$

where $\text{Err}_k = \sum_{i \in C_k} I\{y_i \neq \hat{y}_i\}$ and $\hat{y}_i$ is the predicted class for observation i, obtained from the estimated model with part k removed

- Suppose we are trying to fit a flexible logistic model

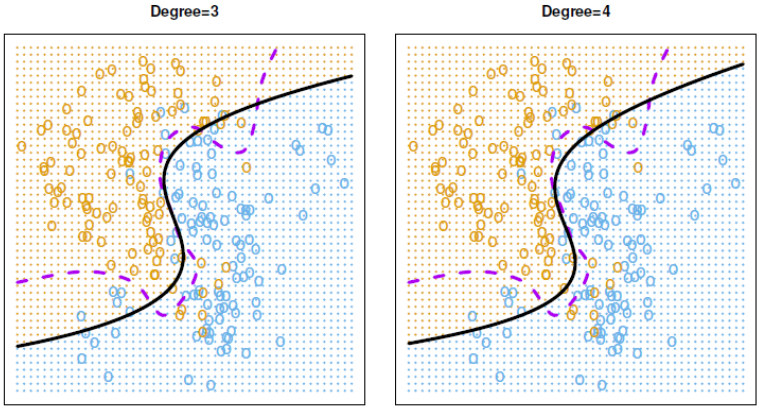
$$\log \frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$$

Classification decision boundaries and true test error rate



Actual test error rate for degree 1 = 0.20 and degree 2 = 0.197

Classification decision boundaries and true test error rate



Actual test error rate for degree 3 = 0.16 and degree 4 = 0.162

K-fold CV for classification

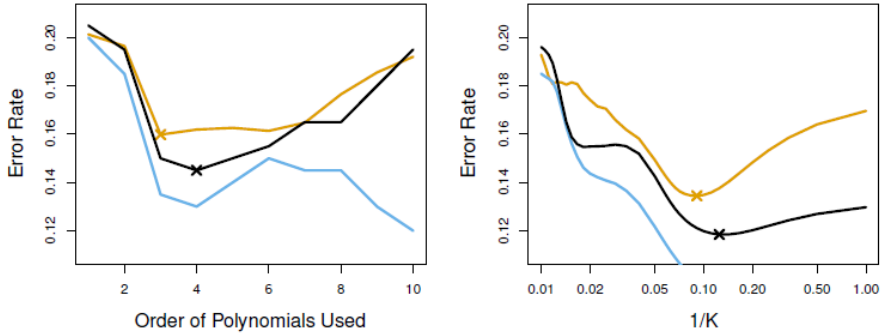


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

KNN