Training Error versus Test error

- Recall the distinction between the *test error* and the *training 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.

- But the 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

resaltar $\rightarrow Y = f(x)$ datos

Descubrimos bien la func. & los parametros cambian.

High Bias

el Modo no Captura la función appropiadamente

Low Variance

Cuando Cambio mi set de entrenamiento el Método no Cambia Mucho

Low Bias

High Variance

Prediction Error

examinación

Test Sample

Training Sample

& entrenamiento

Low

Lineal: (pocos parametros)

Model Complexity

& Complejidad del Método de estimacion.

High red neuronal (muchos parametros)

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
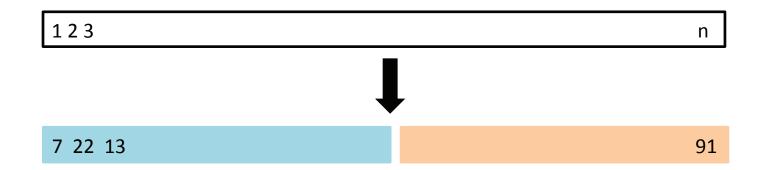
Validation-set approach

- Here 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 in the case of a qualitative (discrete) response.
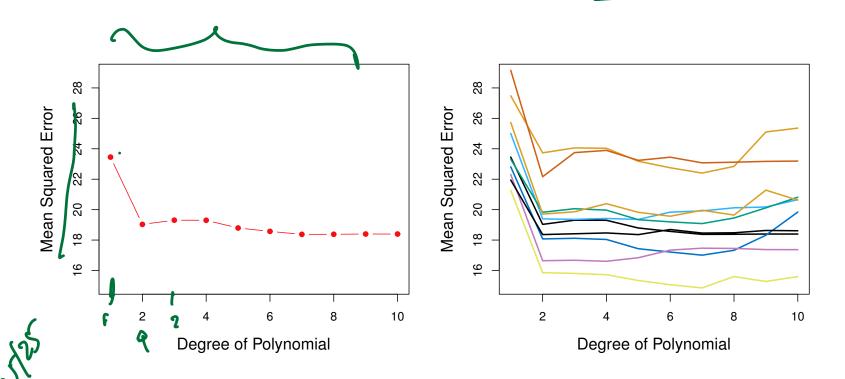
The Validation process

Regresion Logistica.

A) Vsa 100% de los datos para entrenar

B) 50% de los datos para entren

1 2 3 ... n

7 22 13 ... 91

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

EPE (A) ☐ EPE(B)

$\gtrless$

Example: automobile data

- Want to compare linear vs higher-order polynomial terms in a linear regression
- 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

Variable en el set de validacion.

- 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.

  Solo usan algunas de nuestras observacion

- This suggests that the validation set error may tend to *overestimate* the test error for the model fit on the entire data set.

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 data set. *Why?*

K-fold Cross-validation

- *Widely used 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.

K-fold Cross-validation in detail

Divide data into K roughly equal-sized parts ($K = 5$ here)

1	2	3	4	5
Validation	Train	Train	Train	Train

The details

- Let the K parts be $C_1, C_2, \ldots C_K$, where 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

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

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.

The details

- Let the K parts be $C_1, C_2, \ldots C_K$, where 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

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

  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).

A nice special case!

- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

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

  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; see book for details.) This is like the ordinary MSE, except the ith residual is divided by $1 - h_i$.
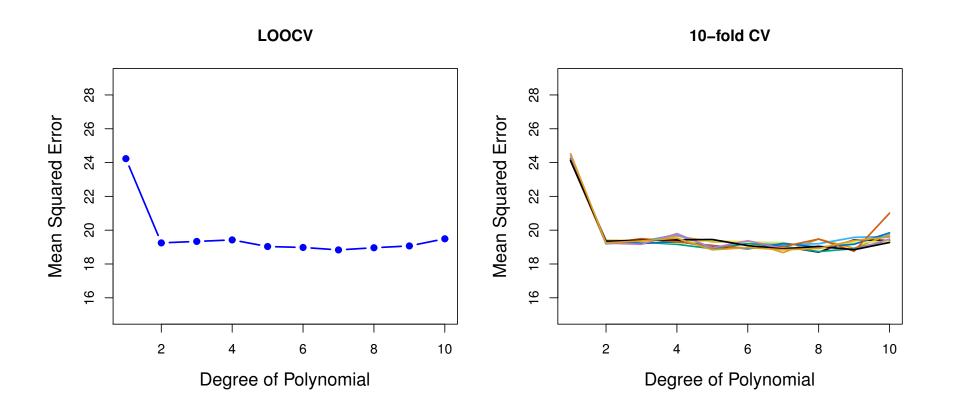
A nice special case!

- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2,$$
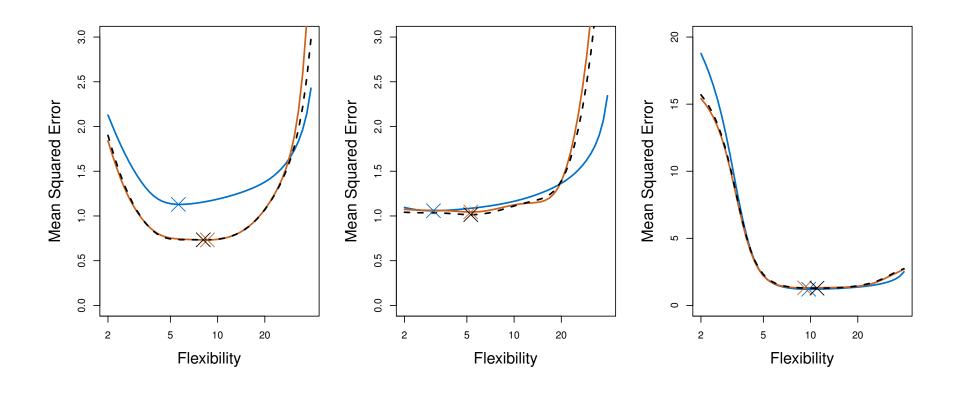
  where $\hat{y}_i$ is the ith fitted value from the original least squares fit, and h_i is the leverage (diagonal of the "hat" matrix; see book for details.) This is like the ordinary MSE, except the ith residual is divided by $1 - h_i$.

- LOOCV sometimes useful, but typically doesn't *shake up* the data enough. 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 revisited

True and estimated test MSE for the simulated data

Other issues with Cross-validation

- 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.

Other issues with Cross-validation

- 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. *Why?*

Other issues with Cross-validation

- 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. *Why?*

- This bias is minimized when $K = n$ (LOOCV), but this estimate has high variance, as noted earlier.

- $K = 5$ or 10 provides a good compromise for this bias-variance tradeoff.

Cross-Validation for Classification Problems

- 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

$$\mathrm{CV}_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$$

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

$$\widehat{\mathrm{SE}}(\mathrm{CV}_K) = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \frac{(\mathrm{Err}_k - \overline{\mathrm{Err}_k})^2}{K-1}}$$

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

Cross-Validation for Classification Problems

- 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

$$\mathrm{CV}_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$$

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

$$\widehat{\mathrm{SE}}(\mathrm{CV}_K) = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \frac{(\mathrm{Err}_k - \overline{\mathrm{Err}_k})^2}{K-1}}$$

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