Cross-validation and the Bootstrap

***

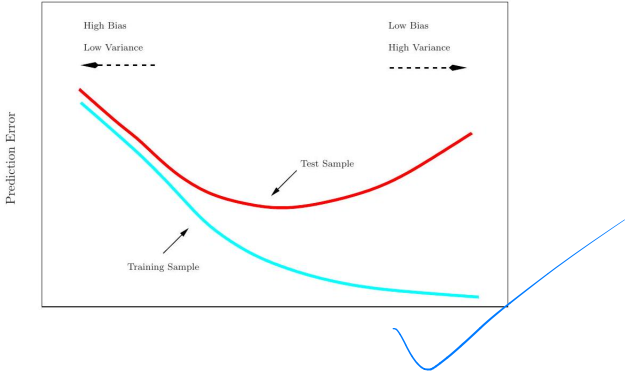
6th March 2023

- In the section 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.
- For example, they provide estimates of test-set prediction error, and the standard deviation and bias of our parameter estimates

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



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

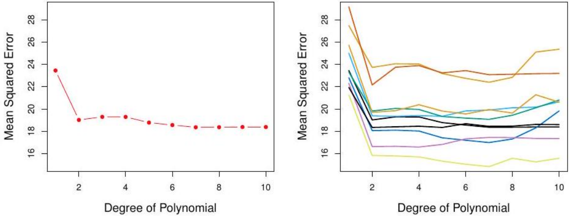
Validation-set approach

- 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

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

- 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 k th 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

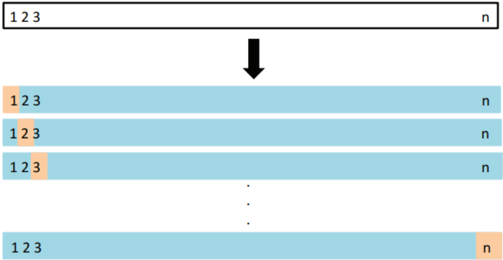
- Let the K parts be $C_1, C_2, \ldots C_K$, where C_k denotes the indices of the observations in part k. There are n_k observations in part k : if N is a multiple of K, then $n_k = n/K$
- Compute

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

where $\mathrm{MSE}_k = \sum_{i \in C_k} \left(y_i - \hat{y}_i\right)^2 / n_k$, 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!



A nice special case!

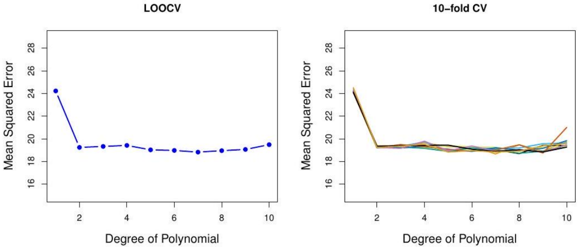
- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

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

$\hat{y}_i$ is the i th fitted value from the original least squares fit, and h_i is the leverage. This is like the ordinary MSE, except the ith residual is divided by $1 - h_i$.

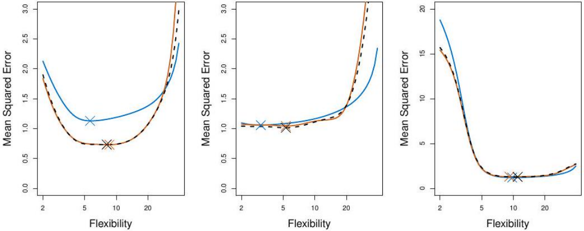
- LOOCV sometimes useful, but typically doesn't shake up the data enough. The estimates from each fold are highly correlated and hence their average can have high variance.
- a better choice is $K = 5$ or 10 .

Auto data revisited



Left: The LOOCV error curve. Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.

True and estimated test MSE for the simulated data



True and estimated test MSE for the simulated data sets. 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

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. LOOCV can be slow when n is large.
- $K = 5$ or 10 provides a good compromise for this bias-variance tradeoff. faster than LOOCV.

Cross-Validation for Classification Problems

- We divide the data into K roughly equal-sized parts $C_1, C_2, \ldots C_K \cdot C_k$ 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 $\mathrm{Err}_k = \sum_{i \in C_k} I\left(y_i \neq \hat{y}_i\right) / n_k$.

- The estimated standard deviation of CV_K is

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

Cross-validation: 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, forgetting about step 1 ?
- NO!

- 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! Try to do this yourself
- We have seen this error made in many high profile genomics papers.

The Wrong and Right Way

- Wrong: Apply cross-validation in step 2.
- Right: Apply cross-validation to steps 1 and 2.

The Bootstrap X X X

- 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.
- For example, 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.

A simple example

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.
- We wish to choose α to minimize the total risk, or variance, of our investment. In other words, we want to $\text{minimize}\, \mathrm{Var}(\alpha X + (1 - \alpha)Y)$.
- One can show that the value that minimizes the risk is given by

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

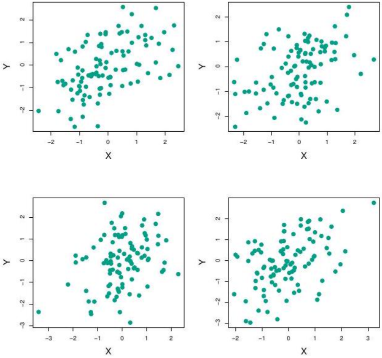
where $\sigma_X^2 = \mathrm{Var}(X), \sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

Example continued

- But the values of σ_X^2, σ_Y^2, and σ_{XY} are unknown.
- We can compute estimates for these quantities, $\hat{\sigma}_X^2, \hat{\sigma}_Y^2$, and $\hat{\sigma}_{XY}$, using a data set that contains measurements for X and Y.
- We can then estimate the value of α that minimizes the variance of our investment using

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

Example continued



Each panel displays 100 simulated returns for investments X and Y. From left to right and top to bottom, the resulting estimates for α are $0.576, 0.532, 0.657$, and 0.651.

Example continued

- To estimate the standard deviation of $\hat{\alpha}$, we repeated the process of simulating 100 paired observations of X and Y, and estimating α 1,000 times.
- We thereby obtained 1,000 estimates for α, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.
- The left-hand panel of the Figure on slide 29 displays a histogram of the resulting estimates.
- For these simulations the parameters were set to $\sigma_X^2 = 1, \sigma_Y^2 = 1.25$, and $\sigma_{XY} = 0.5$, and so we know that the true value of α is 0.6 (indicated by the red line).

Example continued

- The mean over all 1,000 estimates for α is

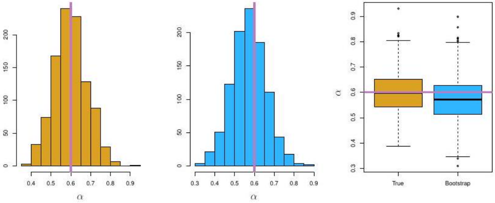
$$\bar{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996$$

very close to $\alpha = 0.6$, and the standard deviation of the estimates is

$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

- This gives us a very good idea of the accuracy of $\hat{\alpha}$: $\mathrm{SE}(\hat{\alpha}) \approx 0.083$
- So roughly speaking, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by

Results

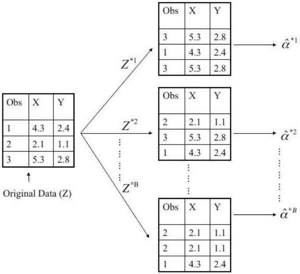


Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population. Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set. Right: The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.

Now back to the real world

- The procedure outlined above cannot be applied, because for real data we cannot generate new samples from the original population.
- However, the bootstrap approach allows us to use a computer to mimic the 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 just 3 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

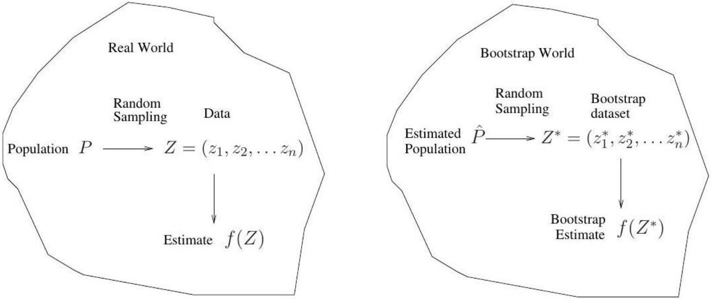
A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α - Denoting the first bootstrap data set 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 100 or 1000), in order to produce B different bootstrap data sets, $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$, 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

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

- This serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set. See center and right panels of Figure on slide 29. Bootstrap results are in blue. For this example $\mathrm{SE}_B(\hat{\alpha}) = 0.087$.

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.
- Also provides approximate confidence intervals for a population parameter. For example, looking at the histogram in the middle panel of the Figure on slide 29 , the 5% and 95% quantiles of the 1000 values is $(.43, .72)$.
- This represents an approximate 90% confidence interval for the true α. How do we interpret this confidence interval?
- The above interval is called a Bootstrap Percentile confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Can the bootstrap estimate 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. Why?
- 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. Can you prove this?
- This will cause the bootstrap to seriously underestimate the true prediction error. Why?
- The other way around- with original sample $=$ training sample, bootstrap dataset $=$ validation sample - is 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. Eg 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.

Some theoretical results for CV

Assume we have a function f which gets the i-th input $\boldsymbol{x}_i$ and outputs $f_i = f(\boldsymbol{x}_i)$. The output may be corrupted with an additive noise ε_i :

$$y_i = f_i + \varepsilon_i$$

where the noise is $\varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right)$. Therefore:

$$\mathbb{E}\left(\varepsilon_i\right) = 0, \quad \mathbb{E}\left(\varepsilon_i^2\right) = \mathrm{Var}\left(\varepsilon_i\right) + \left(\mathbb{E}\left(\varepsilon_i\right)\right)^2 = \sigma^2,$$

The true observation f_i is not random, thus:

$$\mathbb{E}\left(f_i\right) = f_i$$

The input training data $\{\boldsymbol{x}_i\}_{i=1}^n$ and their corrupted observations $\{y_i\}_{i=1}^n$ are available to us. We would like to approximate (estimate) the true model by a model $\widehat{f}$ in order to estimate the observations $\{y_i\}_{i=1}^n$ from the input $\{\boldsymbol{x}_i\}_{i=1}^n$.

Suppose we have an instance $(\boldsymbol{x}_0, y_0)$. This instance can be either a training or test/validation instance. We will cover both cases. The observation y_0 is:

$$y_0 = f_0 + \varepsilon_0.$$

Assume the model's estimation of y_0 is $\widehat{f_0}$. The MSE of the estimation is:

$$\begin{aligned}
\mathbb{E}\left(\left(\widehat{f_0} - y_0\right)^2\right) =&\mathbb{E}\left(\left(\widehat{f_0} - f_0 - \varepsilon_0\right)^2\right) \\
=& \mathbb{E}\left(\left(\widehat{f_0} - f_0\right)^2 + \varepsilon_0^2 - 2\varepsilon_0\left(\widehat{f_0} - f_0\right)\right) \\
=& \mathbb{E}\left(\left(\widehat{f_0} - f_0\right)^2\right) + \mathbb{E}\left(\varepsilon_0^2\right) - 2\mathbb{E}\left(\varepsilon_0\left(\widehat{f_0} - f_0\right)\right) \\
=&\mathbb{E}\left(\left(\widehat{f_0} - f_0\right)^2\right) + \sigma^2 - 2\mathbb{E}\left(\varepsilon_0\left(\widehat{f_0} - f_0\right)\right).
\end{aligned}$$

The last term is:

$$\mathbb{E}\left(\varepsilon_0\left(\widehat{f}_0 - f_0\right)\right) = \mathbb{E}\left((y_0 - f_0)\left(\widehat{f}_0 - f_0\right)\right).$$

For calculation of this term, we have two cases: (I) whether the instance $(\boldsymbol{x}_0, y_0)$ is in the training set or (II) not in the training set. In other words, whether the instance was used to train the model (estimator) or not.

$(\boldsymbol{x}_0, y_0)$ was not in the training set

- Assume the instance $(\boldsymbol{x}_0, y_0)$ was not in the training set, i.e., it was not used for training the model.
- In other words, we have $y_0 \notin \mathcal{T}$. This means that the estimation $\widehat{f}_0$ is independent of the observation y_0 because the observation was not used to train the model but the estimation is obtained from the model.
- Therefore

$$
\begin{aligned}
\therefore \quad & y_0 \perp \widehat{f}_0 \Longrightarrow (y_0 - f_0)\left(\widehat{f}_0 - f_0\right) \\
& \Longrightarrow \mathbb{E}\left((y_0 - f_0)\left(\widehat{f}_0 - f_0\right)\right) \\
& \overset{(a)}{=} \mathbb{E}\left((y_0 - f_0)\right)\mathbb{E}\left(\left(\widehat{f}_0 - f_0\right)\right) \overset{(b)}{=} 0 \times \mathbb{E}\left(\left(\widehat{f}_0 - f_0\right)\right) = 0
\end{aligned}
$$

$(\boldsymbol{x}_0, y_0)$ was not in the training set

where (a) is because $(y_0 - f_0)\left(\widehat{f}_0 - f_0\right)$ and (b) is because:

$$\mathbb{E}\left((y_0 - f_0)\right) = \mathbb{E}\left(y_0\right) - \mathbb{E}\left(f_0\right) \overset{(c)}{=} f_0 - f_0 = 0,$$

where (c) is because of:

$$\mathbb{E}\left(y_0\right) = \mathbb{E}\left(f_0\right) + \mathbb{E}\left(\varepsilon_0\right) = f_0 + 0 = f_0$$

Therefore, in this case, the last term is zero. Thus:

$$\mathbb{E}\left(\left(\widehat{f}_0 - y_0\right)^2\right) = \mathbb{E}\left(\left(\widehat{f}_0 - f_0\right)^2\right) + \sigma^2$$

Suppose the number of instances which are not in the training set is m. We can sum the MSE over all the m instances:

$$\sum_{i=1}^{m}\left(\widehat{f}_i - y_i\right)^2 = \sum_{i=1}^{m}\left(\widehat{f}_0 - f_0\right)^2 + \underbrace{\sum_{i=1}^{m}\sigma^2}_{}.$$

The first term, $\sum_{i=1}^{m}\left(\widehat{f}_i - y_i\right)^2$, is the error for the training data. This error is referred to as empirical error or training error and is denoted by **err**.

The second term, $\sum_{i=1}^{m}\left(\widehat{f}_0 - f_0\right)^2$, is the error for the testing data. This error is referred to as true error or test error and is denoted by **Err**. Therefore:

$$\mathbf{Err} = \mathbf{err} - m\sigma^2.$$

Thus, we can minimize the empirical error in order to properly minimize the true error.

Instance in the training set

Consider a multivariate random variable $\mathbb{R}^d \ni \boldsymbol{z} = [z_1, \ldots, z_d]^\top$ whose components are independent random variables with normal distribution, i.e., $z_i \sim \mathcal{N}(\mu_i, \sigma)$. Take $\mathbb{R}^d \ni \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^\top$ and let $\mathbb{R}^d \ni \boldsymbol{g}(\boldsymbol{z}) = [g_1, \ldots, g_d]^\top$ be a function of the random variable $\boldsymbol{z}$ with $\boldsymbol{g}(\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^d$. There exists a lemma, named Stein's Lemma, which states:

$$\mathbb{E}\left((\boldsymbol{z} - \boldsymbol{\mu})^\top \boldsymbol{g}(\boldsymbol{z})\right) = \sigma^2 \sum_{i=1}^{d} \mathbb{E}\left(\frac{\partial g_i}{\partial z_i}\right)$$

Suppose we take $\varepsilon_0, 0$, and $\widehat{f}_0 - f_0$ as the z, μ, and $g(z)$, respectively. Then

$$\mathbb{E}\left((\varepsilon_0 - 0)\left(\widehat{f}_0 - f_0\right)\right) = \sigma^2 \mathbb{E}\left(\frac{\partial\left(\widehat{f}_0 - f_0\right)}{\partial \varepsilon_0}\right)$$

$$= \sigma^2 \mathbb{E}\left(\frac{\partial \widehat{f}_0}{\partial \varepsilon_0} - \frac{\partial f_0}{\partial \varepsilon_0}\right) \overset{(a)}{=} \sigma^2 \mathbb{E}\left(\frac{\partial \widehat{f}_0}{\partial \varepsilon_0}\right)$$

$$\overset{(b)}{=} \sigma^2 \mathbb{E}\left(\frac{\partial \widehat{f}_0}{\partial y_0} \times \frac{\partial y_0}{\partial \varepsilon_0}\right) \overset{(c)}{=} \sigma^2 \mathbb{E}\left(\frac{\partial \widehat{f}_0}{\partial y_0}\right),$$

where (a) is because the true model f is not dependent on the noise, (b) is because of the chain rule in derivative, and (c) is because:

$$y_0 \overset{(23)}{=} f_0 + \varepsilon_0 \Longrightarrow \frac{\partial y_0}{\partial \varepsilon_0} = 1.$$

Therefore, in this case:

$$\mathbb{E}\left(\left(\widehat{f}_0 - y_0\right)^2\right) = \mathbb{E}\left(\left(\widehat{f}_0 - f_0\right)^2\right) + \sigma^2 - 2\sigma^2 \mathbb{E}\left(\frac{\partial \widehat{f}_0}{\partial y_0}\right).$$

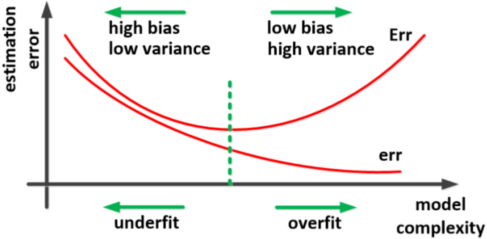
Suppose the number of training instances is n. We can sum the MSE over all the n training instances:

$$\sum_{i=1}^{n}\left(\widehat{f}_i - y_i\right)^2 = \sum_{i=1}^{n}\left(\widehat{f}_0 - f_0\right)^2 + \underbrace{\sum_{i=1}^{n}\sigma^2}_{=n\sigma^2} - 2\sigma^2 \sum_{i=1}^{n}\frac{\partial \widehat{f}_i}{\partial y_i}.$$

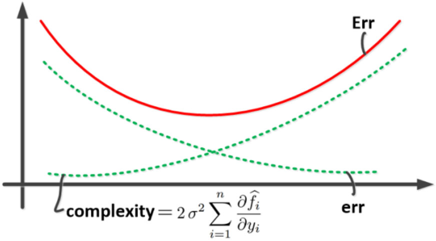
The first term is the empirical error (denoted by err) and the second term is the true error (denoted by Err). Therefore:

$$\mathbf{Err} = \mathbf{err} - n\sigma^2 + 2\sigma^2 \sum_{i=1}^{n}\frac{\partial \widehat{f}_i}{\partial y_i}$$

- The last term is a measure of complexity (or overfitting) of the model.
- Note that $\partial \widehat{f_i} / \partial y_i$ means if we move the i-th training instance, how much the model's estimation of that instance will change? This shows how much the model is complex or overfitted.
- Suppose a line regressing a training set via least squares problem. If we change a point, the line will not change significantly because the model is not complex.
- Consider a regression model passing through "all" the points. If we move a training point, the regressing curve changes noticeably which is because the model is very complex


estimation error
high bias
low variance
low bias
high variance
Err
err
model complexity
underfit
overfit

(a)


Err
complexity $= 2\,\sigma^2 \sum_{i=1}^{n} \frac{\partial \widehat{f}_i}{\partial y_i}$
err

(b)