SF2935: Modern methods of statistical learning
Lecture 4
Resampling methods: Bootstrap and Cross-Validation

Tatjana Pavlenko

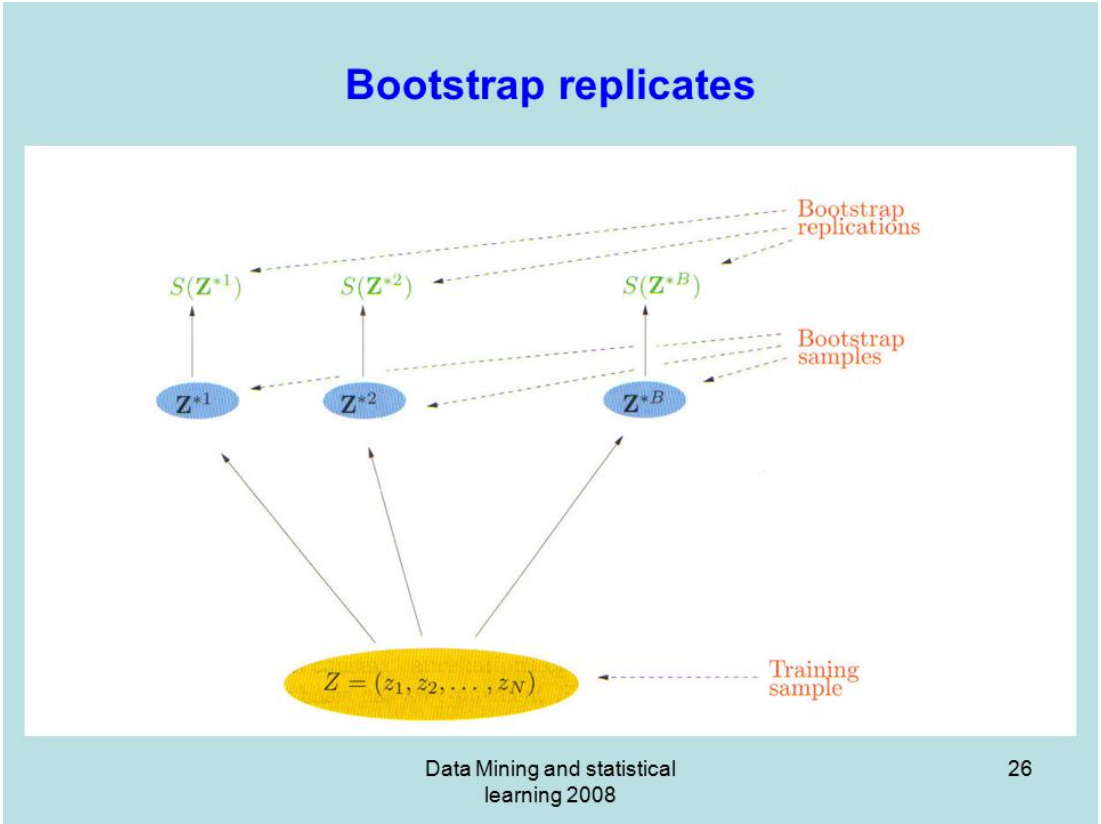
7 November 2017

The bootstrap: basic idea and example

The bootstrap is a general resampling technique for assessing statistical accuracy. We first describe the technique in general.

- Suppose that we have a model fit to a set of training dataset $\mathbf{z} = (z_1, \ldots, z_N)$ where $z_i = (x_i, y_i)$.
- The basic idea of the bootstrap sampling is to randomly draw datasets with replacement from the training data $\mathbf{z}$, each dataset of the same size N as the original dataset. This is done B times, generating B bootstrap datasets.
- We further refit the model to each of the bootstrap sample, and investigate the behavior of the fits over the B replications.

Bootstrap (cont.)



Figur: Schematic of the bootstrap technique. The goal is to evaluate the statistical accuracy of $S(\mathbf{Z})$ computed from the dataset $\mathbf{Z} = (Z_1, \ldots, Z_N)$. B training samples, $\{\mathbf{Z}^{*b}\}_{b=1}^{B}$ each of size N are drawn with replacement from the original dataset. $S(\mathbf{Z})$ is then computed for each bootstrap training sample, and values $\{S(\mathbf{Z}^{*b})\}_{b=1}^{B}$ are used to evaluate the statistical accuracy of $S(\mathbf{Z})$. Obs! These notations are used locally to be consistent with the figure.

THE BOOTSTRAP: BASIC IDEA AND EXAMPLE

Let $S(\mathbf{Z})$ be any quantity (statistics) computed from $\mathbf{Z}$. Using bootstrap sampling we can estimate any aspect of the distribution of $S(\mathbf{Z})$! Assume that we seek to estimate $\mathbb{V}\text{ar}[S(\mathbf{Z})]$.

Bootstrap variance estimator

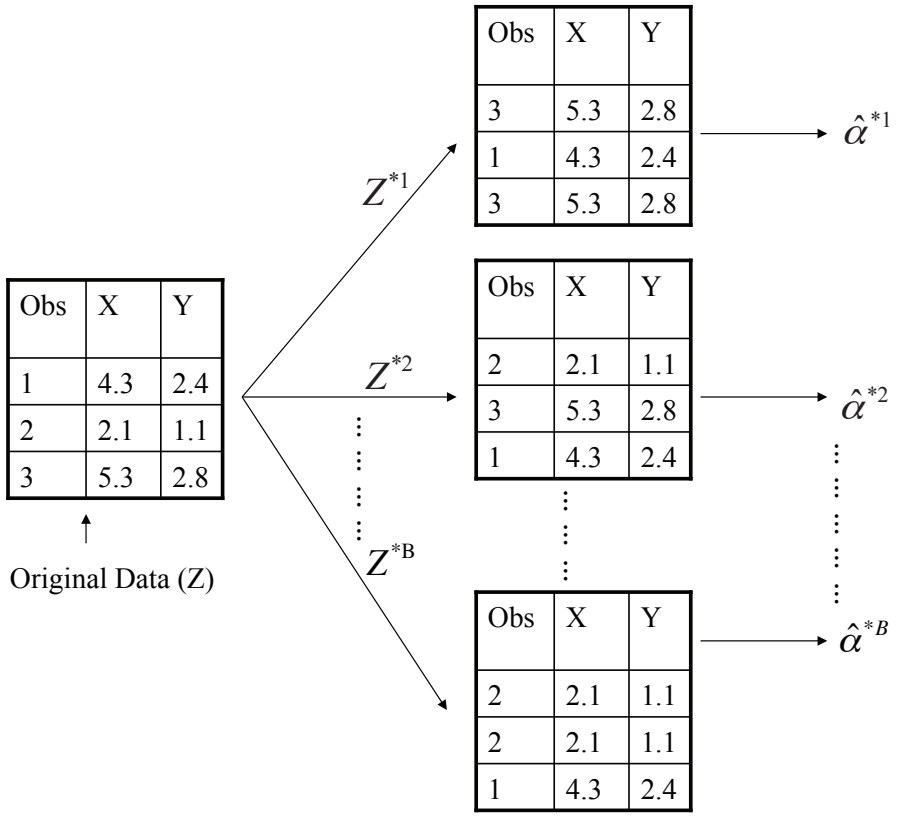
1. Draw a bootstrap sample $\mathbf{Z}^* = (Z_1^*, \ldots, Z_N^*)$. Compute $S(\mathbf{Z}^*)$.
2. Repeat the previous step, B times, yielding samples $\{\mathbf{Z}^{*b}\}_{b=1}^B$ and estimators $S(\mathbf{Z}^{*b})$.
3. Compute

$$\widehat{\mathbb{V}\text{ar}}_{\text{boot}}[S(\mathbf{Z})] = \frac{1}{B-1}\sum_{b=1}^{B}\left(S(\mathbf{Z}^{*b}) - \bar{S}^*\right)^2,$$

where $\bar{S}^* = \sum_{b=1}^B S(\mathbf{Z}^{*b})/B$.

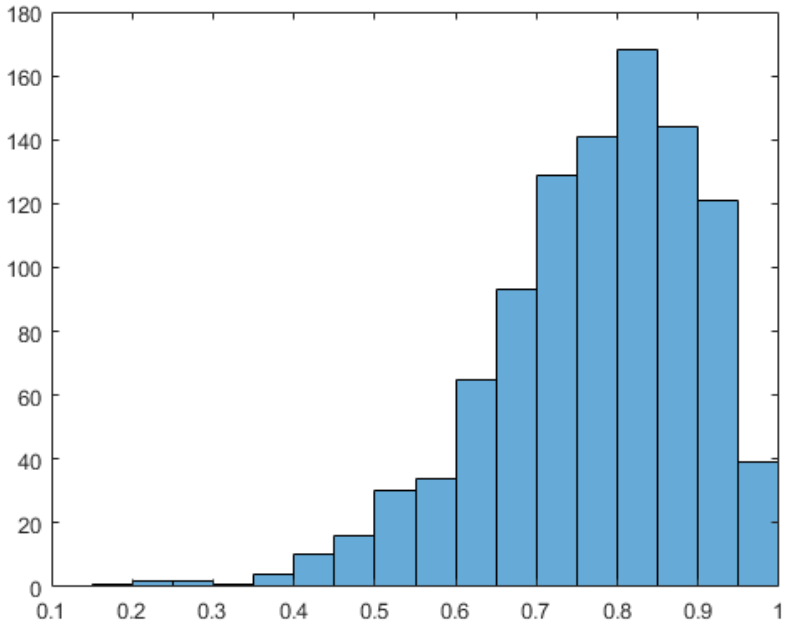
$\widehat{\mathbb{V}\text{ar}}_{\text{boot}}[S(\mathbf{Z})]$ can be thought of as a Monte-Carlo estimate of the variance of $S(\mathbf{Z})$ under sampling form the *empirical distribution function* (EDF) for the data $z = (z_1, \ldots, z_N)$. More details on EDF will be presented later.

Bootstrap methods (cont.)



Figur: In the real world we only get to see the single value of $\hat{\alpha}$, but the bootstrap world is more generous: we can generate as many bootstrap replications $\hat{\alpha}^{*b}$ as we wish (or have capacity for), and directly estimate their variability using e.g. $\widehat{\mathbb{V}\text{ar}}_{\text{boot}}$ or $\widehat{\text{se}}_{\text{boot}}$.

Bootstrap methods (cont.)



Figur: Let α denote the correlation coefficient parameter, i.e $\alpha = Corr(X, Y)$. A histogram of the estimates of α obtained from $B = 1\,000$ bootstrap samples form the original data set. These notations are used locally to be consistent with the figure on slide 5.

BOOTSTRAP(CONT.)

FIGUR: The term bootstrap attributed to a story in *The Surprising Adventures of Baron Münchausen* where the eponymous Baron pulls himself out of a swamp by his own bootstraps. Published in London in 1785 by Rudolf Erich Raspe.

Why does the bootstrap work? Statistical issues

There are two senses in which bootstrap resampling methods might *work:*

- Q1: Do these methods provide reliable results when being used with the data generated in applications?
- Q2: Under what theoretical (idealized) conditions will a bootstrap resampling technique provide results that are in some sense statistically correct?

An important role in answering these questions is played by the empirical distribution function, (EDF).

Assume the following.

- We have a sample of data, $x = (x_1, \ldots, x_n)$ which are thought as the outcomes of iid r. v. $X_1, \ldots, X_n$ whose pdf and cdf are f and F, respectively.
- The sample is to be used to make inference about a population characteristic (for ex. parameter of F), denoted by θ ($\theta = \theta(F)$ which means that θ is some function of the true unknown distribution F) using a statistics $t = t(X)$ whose value at the sample is $t(x)$.
- The choice of t has been made, i.e it is an estimator of θ which we take to be a scalar. For instance, let θ be the mean of F, then

$$\theta(F) = \mathbb{E}_F(X) = \int xf(x)dx,$$

- We focus attention on the uncertainty of estimator $t(X)$. For example, what is its bias, its standard error $\mathrm{se}(t(X))$ or its quantiles? How do we obtain confidence limits for θ using $t(X)$? All these questions are concerned with the probability distribution of $t(X)$.

The Bootstrap in general (cont.)

In the real world $\widehat{\theta} = t(\boldsymbol{x})$ is obtained in two steps: first $\boldsymbol{x}$ is generated by iid sampling from F and then $\widehat{\theta}$ is calculated from $\boldsymbol{x}$ according to $t(\cdot)$, i.e

$$F \xrightarrow{\text{iid}} \boldsymbol{x} \xrightarrow{t(\cdot)} \widehat{\theta}.$$

Since we don't know F the natural way to proceed is

- replace F with its data-based approximation denoted by $\widehat{F}$.
- use the Monte-Carlo sampling from the approximation $\widehat{F}$ to investigate the variation in $t(X)$.

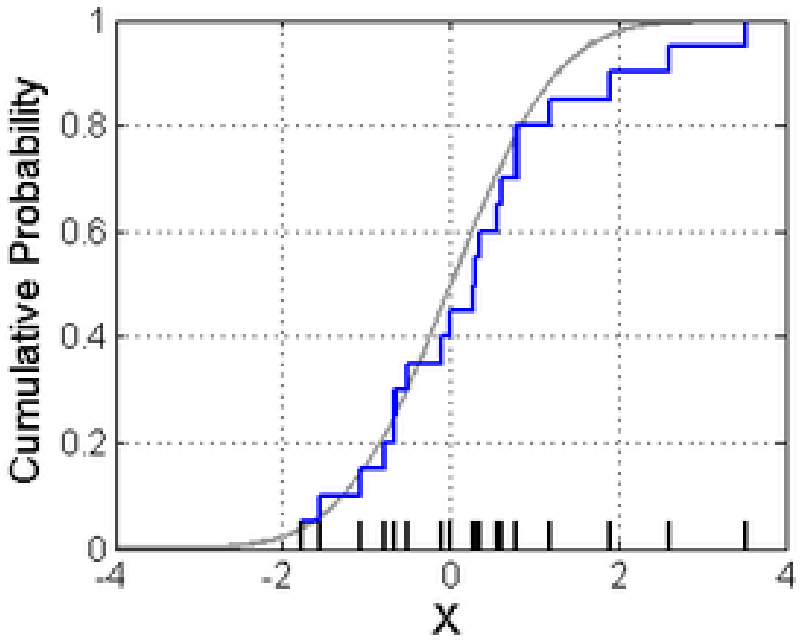
The common strategy to obtain the approximation $\widehat{F}$ is to use the empirical distribution function (EDF).

- The EDF associated with the dataset $\boldsymbol{x} = (x_1, \ldots, x_n)$ is defined as a sample proportion

$$\widehat{F}_n(u) = \frac{\#\{x_i \leq u\}}{n} = \frac{1}{n}\sum_{i=1}^{n} H(u - x_i),$$

$H(u)$ is the unit step function which jumps from 0 to 1 at $u = 0$.

Bootstrap methods (cont.)



Figur: The blue line represents an empirical distribution function and the gray curve is the true cumulative distribution function. Notice that the values of EDF are fixed at $(0, \frac{1}{n}, \frac{2}{n} \ldots, \frac{n}{n})$, so the EDF is equivalent to its points of increase, the ordered values $(x_{(1)}, \ldots, x_{(n)})$ of the data.

The bootstrap sampling and EDF

$\widehat{F}_n(x)$ plays the role of fitted population model when no mathematical form is assumed for F. How to sample from $\widehat{F}_n(x)$?

- Observe that $\widehat{F}_n(u) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{x_i \le u\}}$ of $\boldsymbol{x} = (x_1, \ldots, x_n)$ places equal weights of $1/n$ to each distinct observed value x_i.
- This implies that a bootstrap sample $\mathbf{x}^*$ *is an iid sample drawn from* $\widehat{F}_n$. Then each x^* independently has equal probability of $1/n$ of being any member of $\{x_1, \ldots, x_n\}$
- This in turn implies that a sample $\boldsymbol{x}^*$ of size n from the empirical distribution $\widehat{F}_n(x)$ associated with the data $\boldsymbol{x} = (x_1, \ldots, x_n)$ can be generated as follows:
  - sample independently n indices $\imath_1, \imath_2 \ldots, \imath_n$ form the uniform distribution over the set of integers $\{1, 2 \ldots, n\}$;
  - let $\boldsymbol{x}^* = (x_{\imath_1}, x_{\imath_2}, \ldots, x_{\imath_n})$.
- This sampling procedure draws n elements from the set $\{x_1, x_2, \ldots, x_n\}$ with replacement! (Recall the bootstrap.)

Some properties of $\widehat{F}_n$.

- For a fixed u, $\mathbb{1}_{\{X_i \le u\}}$ is $Be(p)$ random variable with $p = F(u)$. Hence $n\widehat{F}_n(u) = \sum_{i=1}^n \mathbb{1}_{\{X_i \le u\}}$ is $Bin(n, F(u))$. This implies that $\widehat{F}_n(u)$ is an unbiased estimator of $F(u)$.
- By the *strong law of large numbers*, as $n \to \infty$

$$\widehat{F}_n(u) \xrightarrow{a.s.} F(u),$$

i.e. $\widehat{F}_n(u)$ converges to $F(u)$ almost surely, for every value of u. Thus $\widehat{F}_n(u)$ is a consistent estimator of the CDF $F(u)$.

- By CLT, point-wise, $\widehat{F}_n(u)$ has asymptotically normal distribution as $n \to \infty$

$$\sqrt{n}\left(\widehat{F}_n(u) - F(u)\right) \xrightarrow{D} N(0, F(u)(1 - F(u)),$$

with the standard $\sqrt{n}$ rate of convergence.

.... OF $\widehat{F}_n$.

- Now having observed data $\boldsymbol{x} = (x_1, \ldots, x_n)$ and using the asymptotic properties of $\widehat{F}_n$ we may *replace* F with $\widehat{F}_n$ in such a way that any quantity involving F can now be approximated by plugging $\widehat{F}_n$ into the quantity instead as $\theta = \theta(F) \approx \widehat{\theta} = \theta(\widehat{F}_n)$.
  For example if θ is the mean of F then

$$\theta = \mathbb{E}_F(X) \approx \widehat{\theta} = \mathbb{E}_{\widehat{F}_n}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

- It can be shown that $\widehat{F}_n$ maximizes the probability of obtaining the observed sample $\boldsymbol{x}$ under all possible choices of F in $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ given that we have observed $x_i \overset{iid}{\sim} F$, i.e it is the nonparametric MLE of F.

Bootstrap estimates

Q: How to investigate the uncertainty in estimator $t(\boldsymbol{X})$ of θ using bootstrap?

The uncertainty of $t(\boldsymbol{X})$ is expressed in terms of the *error distribution*, i.e. the distribution of $t(\boldsymbol{X}) - \theta$. We study the error distribution by considering its approximation $t(\boldsymbol{X}) - \widehat{\theta}$ where the observed value of $\widehat{\theta} = \widehat{\theta}(\boldsymbol{x})$ replaces the unknown θ. (The error distribution is thus a shift of the $t(\boldsymbol{X})$ distribution.)

Bootstrap estimated error distribution

1. Construct $\widehat{F}_n$ using $\boldsymbol{x} = (x_1, \ldots, x_n)$.
2*. Draw a sample $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ from $\widehat{F}_n$ and compute $t(\mathbf{x}^*) - \widehat{\theta}$.
3. Repeat the previous step, B times, yielding samples $\{\mathbf{x}^{*b}\}_{b=1}^B$ and corresponding set of bootstrap estimates $\{t(\mathbf{x}^{*b}) - \widehat{\theta}\}_{b=1}^B$.
4. The error distribution is estimated by considering the sample distribution of the $\{t(\mathbf{x}^{*b}) - \widehat{\theta}\}_{b=1}^B$ (for ex by a histogram).

Obs! In the case of the empirical distribution, sampling from $\widehat{F}_n$ at step 2*. is the same as drawing, with replacement from the dataset $\boldsymbol{x} = (x_1, \ldots, x_n)$.

Bootstrap estimates (cont.)

- Let F_ε be the error distribution associated with $\varepsilon = t(\boldsymbol{X}) - \theta$. A (equi-tailed) confidence set I_θ for θ with the coverage probability $1 - \alpha$ is then defined by the limits

$$I_\theta = \left(\widehat{\theta} - F_\varepsilon^{-1}(1 - \alpha/2), \widehat{\theta} - F_\varepsilon^{-1}(\alpha/2)\right).$$

  A confidence set satisfies $P(\theta \in I_\theta) = 1 - \alpha$ exactly, but it not possible to specify such sets except in a few special parametric models.

- Bootstrap sampling algorithms can be applied to obtain confidence sets as follows:
- The sample of bootstrapped errors $\{\varepsilon_b^*\}_{b=1}^B = \{t(\mathbf{x}^{*b}) - \widehat{\theta}\}_{b=1}^B$ is approximately distributed as F_ε and can be used to approximate the quantiles of F_ε.

Bootstrap confidence set

- The sample of bootstrapped errors $\{\varepsilon_b^*\}_{b=1}^B = \{t(\mathbf{x}^{*b}) - \widehat{\theta}\}_{b=1}^B$ is approximately distributed as F_ε and can be used to obtain estimated quantiles of F_ε.
- The B simulated values $\{\varepsilon_b^*\}_{b=1}^B$ are ordered as $(\varepsilon_{(1)}^*, \dots \varepsilon_{(B)}^*)$ and the α-quantile of F_ε is estimated by the $(B+1)\alpha$th of these, i.e. for any $\alpha \in (0,1)$, $\quad F_\varepsilon^{-1}(p) \approx \varepsilon_{((B+1)\alpha)}^*$.
- The confidence limits of I_θ are then replaced by their bootstrap based counterparts which gives $1-\alpha$ bootstrap confidence set for θ

$$I_\theta^B = \left(\widehat{\theta} - \varepsilon_{((B+1)(1-\alpha/2))}^*, \widehat{\theta} - \varepsilon_{((B+1)(\alpha/2))}^*\right).$$

- The bootstrap estimation of the confidence set considered above is *non-parametric* in the sense that no model is assumed for the distribution F underlying the data.
- Sampling strategy was

$$\widehat{F}_n \xrightarrow{\text{iid}} \{\mathbf{x}^{*b}\}_{b=1}^{B} \longrightarrow \{t(\mathbf{x}^{*b})\}_{b=1}^{B},$$

  (from the empirical distribution of the real data), i.e. drawings form the data with replacement.
- Sampling from the empirical distribution $\widehat{F}_n$ can be treated as sampling from a finite population, where the *population* is the observed data $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. This process is therefore called for *resampling*. To be iid sampling has to be with replacement.
- The bootstrap methods work well when $\widehat{F}_n$ is close to the true unknown distribution F . This will usually be the case when the sample size n is large enough.

The parametric bootstrap

- In the non-parametric bootstrap, the distribution F underlying the data is assumed to be completely unknown (only iid data).
- In the parametric bootstrap, we assume that data $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ are generated as iid from a parametric family F_θ with $\theta \in \Theta$. Then to utilize the known structure of F we
  - first obtain an estimator of θ, $\widehat{\theta} = \widehat{\theta}(\boldsymbol{x})$ ($\widehat{\theta}$ should be a good estimator of θ, for example MLE),
  - and draw new bootstrap samples $\{\mathbf{x}^{*b}\}_{b=1}^{B}$ from the (plug-in) distribution $\widehat{F} = F(\widehat{\theta})$, from which the usual bootstrap estimators $\{t(\mathbf{x}^{*b})\}_{b=1}^{B}$ are then constructed.
- The resamplig strategy is

$$F(\widehat{\theta}) \longrightarrow \{\mathbf{x}^{*b}\}_{b=1}^{B} \longrightarrow \widehat{\theta}^*$$

and proceed as before to calculate e.g. $\widehat{\mathrm{se}}_{\mathrm{boot}}$.

Bootstrapping in regression

Bootstrap technique can be used for estimating prediction error in regression models. The specific version of bootstrap to be applied depends on the assumption on stochastic model generating the data. Specifically, whether random-$\boldsymbol{X}$ or fixed-$\boldsymbol{X}$ case is considered.

Assume the following:

- The learning data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$ are iid observations from the joint probability distribution of $(\boldsymbol{X}, Y)$ (in $p+1$ dimensional space) that are generated by the regression model,

$$Y = \beta_0 + \boldsymbol{X}'\beta + \varepsilon = \mu(\boldsymbol{X}) + \varepsilon,$$

  $\mu(\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X}) = \beta_0 + \boldsymbol{X}'\beta$ is the true regression function, $\mathbb{E}(\varepsilon|\boldsymbol{X}) = 0$ and $\mathbb{V}\mathrm{ar}(\varepsilon|\boldsymbol{X}) = \sigma^2$.

- Given $\mathcal{D}$ we can estimate prediction error, PE by

$$\widehat{PE}(\hat{\mu}, \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}(\boldsymbol{x}_i))^2) = \frac{RSS}{n}$$

Bootstrapping in regression (cont.)

- **Obs!** $\widehat{PE}(\hat{\mu}, \mathcal{D})$ is *apparent error rare* or resubstitution error rate for $\mathcal{D}$. $\widehat{PE}(\hat{\mu}, \mathcal{D})$ is computed by fitting OLS regression using the original data $\mathcal{D}$ and then by applying that function to see how well it predicts those same members of $\mathcal{D}$.
- We expect that $\widehat{PE}(\hat{\mu}, \mathcal{D})$ will be too optimistic to estimate PE, i.e. $\widehat{PE}(\hat{\mu}, \mathcal{D}) < PE$.
- Q: How to improve the estimation of the prediction error?
- Computationally intensive correction strategy, i.e the bootstrap, that can be used to improve $\widehat{PE}(\hat{\mu}, \mathcal{D})$.
- **Obs!** With the joint distribution of $(\boldsymbol{X}, Y)$ the resampling involves sampling pairs with replacement form $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$.

KTH Matematik

Bootstrap estimated prediction error

1. Draw a sample of (pairs) with replacement form $\mathcal{D}$, denote it by $\mathcal{D}^{*B} = \{(\boldsymbol{x}_i^{*b}, y_i^{*b}), i = 1, \ldots, n\}_{b=1}^{B}$.

2. For each b regress y_i^{*b} on $\boldsymbol{x}_i^{*b}$ to obtain OLS regression function $\hat{\mu}^{*b}(\boldsymbol{x})$.

3. Compute a *simple bootstrap estimator* of PE by applying $\hat{\mu}^{*b}$ to the *original* sample $\mathcal{D}$ as

$$\widehat{PE}(\hat{\mu}^{*B}(\boldsymbol{x}), \mathcal{D}) = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n}\left(y_i - \hat{\mu}^{*b}(\boldsymbol{x}_i)\right)^2.$$

4. Apply $\hat{\mu}^{*b}$ to $\mathcal{D}^{*B}$ and compute

$$\widehat{PE}(\hat{\mu}^{*B}(\boldsymbol{x}), \mathcal{D}^{*B}) = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n}\left(y_i^{*b} - \hat{\mu}^{*b}(\boldsymbol{x}_i)\right)^2.$$

5. Compute *the overall optimism*

$$opt^{*B} = \widehat{PE}(\hat{\mu}^{*B}(\boldsymbol{x}), \mathcal{D}) - \widehat{PE}(\hat{\mu}^{*B}(\boldsymbol{x}), \mathcal{D}^{*B})$$

6. Compute the bootstrap bias-corrected estimator of PE as

$$PR^* = \widehat{PE}(\hat{\mu}, \mathcal{D}) + opt^{*B}$$

Bootstrapping in regression: further improvements

- Observe that the simple bootstrap estimator $\widehat{PE}(\hat{\mu}^{*b}(\boldsymbol{x}), \mathcal{D})$ underestimates PE because there are observations *common to the bootstrap samples in $\mathcal{D}^{*B}$ (operating as a learning data) and to the original data $\mathcal{D}$ (operating as the test set)*.
- The probability that the observation ith $(\boldsymbol{x}_i, y_i)$ from $\mathcal{D}$, is selected at least once to be in the bth bootstrap sample $\mathcal{D}^{*b}$ is approximated when $n \to \infty$ as

$$\mathbb{P}\mathrm{r}\left((\boldsymbol{X}_i, Y_i) \in \mathcal{D}^{*b}\right) = 1 - \left(1 - \frac{1}{n}\right)^n \to 1 - e^{-1} \approx 0.632.$$

- On average, about 37% of observations in $\mathcal{D}$ are left out of each bootstrap sample, which contains $0.632n$ distinct observations.
- One consequence: if p is close to n there could singularity or near singularity in $\mathcal{X}'\mathcal{X}$ ($\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$ denotes the $n \times p$ design matrix).

Bootstrapping in regression: further improvements

- Q: How to improve PE^*? Solution: The *leave-one-out bootstrap* strategy. Let $PE_{(1)}$ denote the expected bootstrap prediction error at those observations $(\boldsymbol{x}_i, y_i)$ that are *not* included in the B bootstrap samples.
- In general, estimate $PE_{(i)}$ as $PE^*_{(i)} = \frac{1}{B_i} \sum_{b \in C_i} \left(y_i - \mu^*_b(\boldsymbol{x}_i)\right)^2$ where C_i is the set of indices of the bootstrap samples that do not contain $(\boldsymbol{x}_i, y_i)$, and $B_i = |C_i|$ is the number of such bootstrap samples.
- The *0.632 bootstrap estimator of optimism* is given by

$$opt^{*B}_{0.632} = 0.632(PE^*_{(1)} - \widehat{PE}(\hat{\mu}, \mathcal{D})).$$

- Replacing opt^{*B} with $opt^{*B}_{0.632}$ we obtain *0.632 bootstrap estimator of PE* as

$$PE^*_{0.632} = \widehat{PE}(\hat{\mu}, \mathcal{D}) + opt^{*B}_{0.632} = 0.368 \cdot \frac{RSS}{n} + 0.632 \cdot PE^*_{(1)}.$$

In summary

- Non-parametric bootstrap
  - assumes that the data $\boldsymbol{x} = (x_1, \ldots, x_n)$ are iid observations from an *unknown distribution*, i.e no assumption on F;
  - is the resampling procedure where the simulated sample $\boldsymbol{x}^* = (x_1^*, \ldots, x_n^*)$ is a random sample taken with replacement from the original data.
- Parametric bootstrap
  - assumes that the data $\boldsymbol{x} = (x_1, \ldots, x_n)$ are iid observations from a specific parametric family F_θ.
  - resamples a *known distribution function*, whose parameters are estimated from the original data.

In summary (cont.)

Some advantages and disadvantages of both approaches.

- In the non-parametric bootstrap, samples are drawn from a discrete set of n observations. For small n, might underestimate the amount of variation in the population we originally sampled. Generally, the samples of $n \leq 10$ are too small for obtaining reliable nonparametric bootstrap estimators.
- Parametric bootstrap usually requires less data to obtain reliable estimators due to stronger assumptions but is sensitive to the model choice. EDF (used in the non-parametric bootstrap) based on even quite small samples deviates little from that of its population one.
- ML estimators are commonly used in the parametric bootstrap. However their good properties nearly always are based upon their large sample behaviour.

Read more about bootstrap in e.g *Bootstrap methods and their applications*, by A.C. Davison and D. V. Hinkley.

Cross-Validation: Review of the learning problem

Performance of a learning method naturally relates to its prediction accuracy on *independent* test data. We discuss *Cross-Validation*, CV, a simple and very efficient method for estimating prediction error. Initially derived by Stone (1974), but there are several versions.

- Let Y be a quantitative response variable, $\mathbf{X}$ a vector of feature variables/predictors, and a prediction model $\widehat{f}(\mathbf{X})$, which has been estimated using the training data $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
- The common loss function for measuring errors (discrepancies) between Y and $\widehat{f}(\mathbf{X})$ is the squared error

$$L(Y, \widehat{f}(\mathbf{X})) = (Y - \widehat{f}(\mathbf{X}))^2.$$

- We will be considering various types of estimation of the loss.

Performance assessment: Test and training errors.

- *Test error*, also called for *generalization* error, is defined as

$$\mathrm{Err}_{\mathcal{T}} = \mathbb{E}\left(L(Y, \widehat{f}(\mathbf{X}))|\mathcal{T}\right).$$

  - $(\mathbf{X}, Y)$ is *new*, an independent (on $\mathcal{T}$) data drawn randomly from the joint population distribution of $\mathbf{X}$ and Y, assumed to be the same as for $\mathcal{T}$).
  - The training set $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is fixed.

- *Expected test error*, also known as expected prediction error, is

$$\mathrm{Err} = \mathbb{E}\left[\mathbb{E}\left(L(Y, \widehat{f}(\mathbf{X}))|\mathcal{T}\right)\right],$$

  where the expectation averages over all the randomness, including that in the training set $\mathcal{T}$ which produces the estimated model $\widehat{f}$.

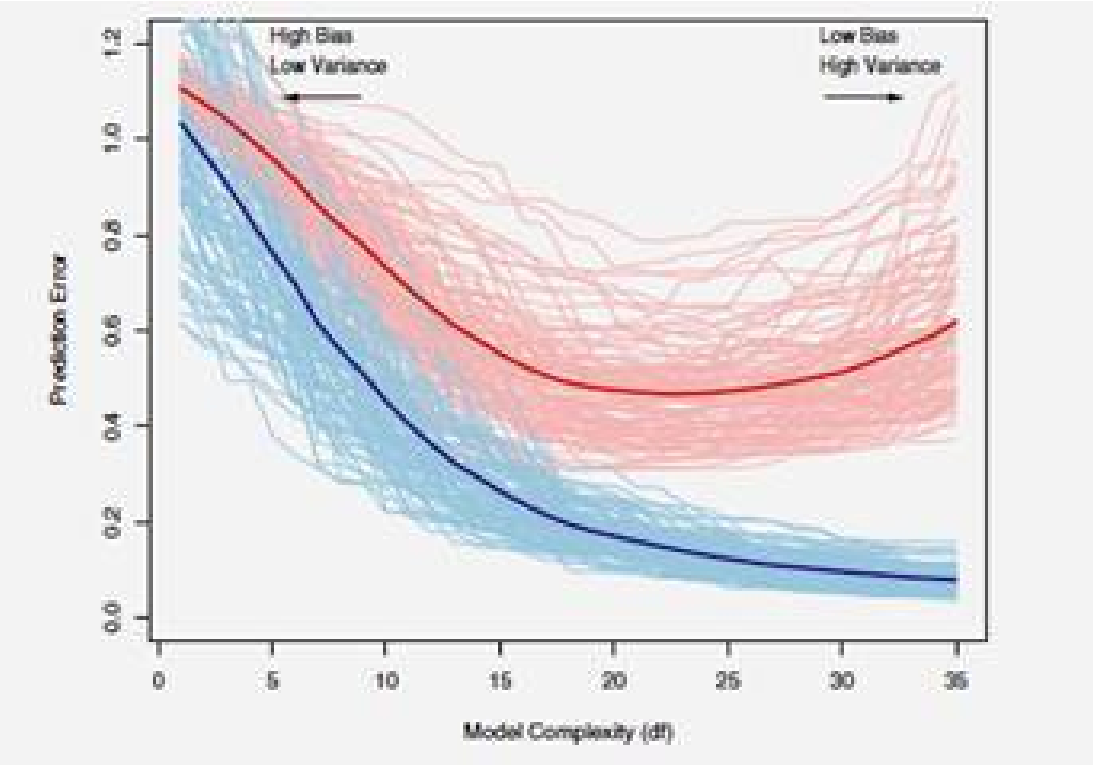
Performance assessment: Test and training errors (cont).

- *Training error* is also known as *apparent* or *re-substitution* error rate, is the average loss over the training sample

$$\overline{\text{err}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \widehat{f}(\mathbf{x}_i)).$$

  - Is the most natural estimate of the error rate and shows how well we predict those *same* members of $\mathcal{T}$.
  - For almost all learning procedures, $\overline{\text{err}}$ is readily available after training.
  - Have misleadingly optimistic value since it estimates the predictive ability of the fitted model from the same $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ that was uses to fit the model! But we are interested in the accuracy of the prediction which is obtained from applying the learned/fitted model to previously unseen test data.
  - Generally, $\overline{\text{err}} \ll \text{Err}_{\mathcal{T}}$.
- This phenomena is presented on the figure below.

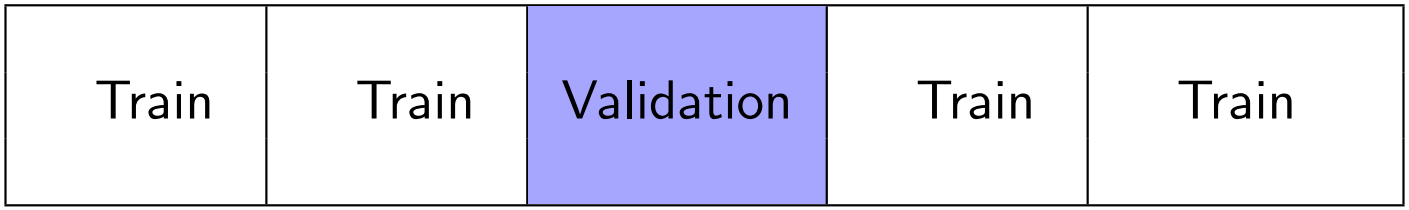
Training- vs test-performance with varying model complexity



Figur: Behavior of the test and training error as a function of model complexity. Light blue curves represent a set of training errors. Solid blue curve represents the expected training error. Light red curves represent conditional test errors. Solid red curve represents the expected test error.

K-Fold Cross-Validation: Generic idea

- Split randomly the set of available data into two parts: a *training set* and a *validation* or *hold-out set*
- For example, split into $K = 5$ roughly equal-size parts:



- Leave out part k (third above), fit model to the other $K - 1$ parts (combined together), specify the prediction error of the model fit when predicting the left-out kth part of the data.
- Perform this for $k = 1, 2, \ldots, K$ and combine the K estimates (usually by averaging over splits) into validation-set error which provides an estimate of the *test error*.
- The beauty of CV is its simplicity and generality:

  it allows for estimation of the test error rate by using (re-using) the training data!

K-Fold Cross-Validation: More precisely

- Let $\kappa : \{1, \ldots, n\} \to \{1, \ldots, K\}$ denote the indexing function that indicates the partition to which observation i is allocated by the randomization.
- Let $\widehat{f}^{-k}(\mathbf{x})$ denote the function fitted with the kth part of the data removed.
- Then the *cross-validation* estimate of the prediction error is given by

$$\mathrm{CV}_{(K)} = \frac{1}{K} \sum_{k=1}^{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_{\kappa(i)}} L(y_i, \widehat{f}^{-\kappa(i)}(\mathbf{x}_i)).$$

- Usual choices for K are 5 or 10.
- The special case of $K = n$ is known as *leave-one-out* cross-validation, LOOCV. In this case, $\kappa(i) = i$, and for the ith observation the fit is computed using all the data except the ith.

K-Fold Cross-Validation: More precisely

- Given a set of models $f(x, \alpha)$ indexed by a tuning parameter α, denote by $\widehat{f}^{-k}(\mathbf{x}, \alpha)$ the αth model fit with the kth part of the data removed.
- For this set of the models define

$$\mathrm{CV}_{(K)}(\alpha) = \frac{1}{K} \sum_{k=1}^{K} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_{\kappa(i)}} L(y_i, \widehat{f}^{-\kappa(i)}(\mathbf{x}_i, \alpha)).$$

- The function $\mathrm{CV}(\alpha)$ provides an estimate of the test error curve; $\hat{\alpha}$, a minimizer of it, is the the (CV) optimal value of the tuning parameter.
- The final chosen model is $f(x, \hat{\alpha})$ which we then fit to *all the data*.

K-Fold Cross-Validation: MSE loss

- Let the K parts be $C_1, \ldots, C_k$ where C_k is the set of indices of the observations in part k, there are n_k obs. in part k.
- Assume that n is a multiple of K, so that $n_k = n/K$ and consider linear fitting under squared-error loss

$$L(\mathbf{y}, \widehat{f}^{-\kappa(i)}(\mathbf{x})) = \text{MSE}^{-k} = \frac{1}{n_k} \sum_{i \in C_k} (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i$ is the fit for observation i obtained from the data with part k removed. Then

$$\text{CV}(\widehat{f}) = \sum_{k=1}^{K} \frac{n_k}{n} \text{MSE}^{-k}$$

- Setting $K = n$ yields n-fold of leave-one out, LOOCV.

CV ON CLASSIFICATION PROBLEM

- Let Y be a qualitative response variable and $\mathbf{X}$ a vector of feature variables/predictors. This corresponds to the classification model with $Y \in \mathcal{C} = \{1, \ldots, c\}$ given classes. The training data set $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. In the classification setting we then assign an observation $\mathbf{x}$ to the class c for which

$$\hat{y}(\mathbf{x}) = \hat{y}_c = \arg\max_{c \in \mathcal{C}} \mathbb{P}(Y = c | \mathbf{X} = \mathbf{x}).$$

- 0-1 loss function, $L(Y, \widehat{Y}(\mathbf{X})) = I(Y \neq \widehat{Y}(\mathbf{X}))$, can be used to quantifying the accuracy of the classifier.
- Then the *training* error is $\overline{\text{err}} = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{y}_i)$ with averaging over $\mathcal{T}$,
- and the *test* error is $\text{Err}_{\mathcal{T}} = E(L(Y, \widehat{Y}(\mathbf{X}) | \mathcal{T})$.
- The test error associated with a *new* set of observations (independent on $\mathcal{T}$!) $\mathcal{T}_{new} = \{(\mathbf{x}_{new}, y_{new})\}$ is estimated by

$$\text{Err} = \text{Ave}(\text{Err}_i) \quad \text{where} \quad \text{Err}_i = I(y_i \neq \hat{y}_i) \text{ for } \mathcal{T}_{new}.$$

CV ON CLASSIFICATION PROBLEM (CONT.)

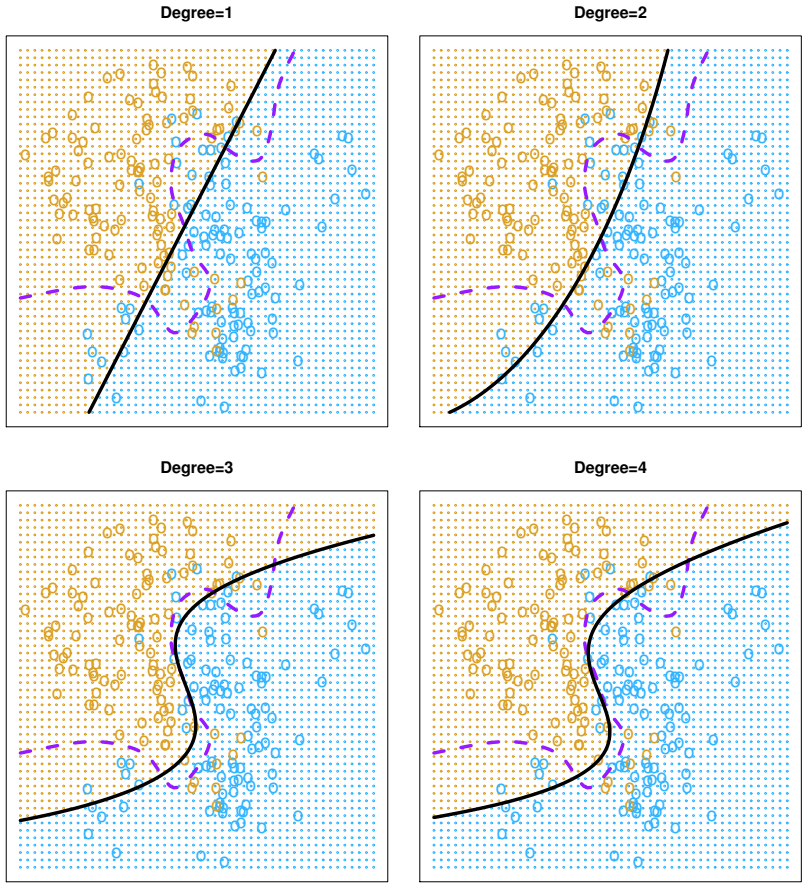
- K-**fold CV**. Split the data $\mathcal{T}$ randomly into K parts, $\mathcal{T}_{C_1}, \ldots, \mathcal{T}_{C_K}$ where C_k is the set of indices of the observations in part k and let $n_k = n/K$.
  - For each k compute $\text{Err}_k = \sum_{i \in \mathcal{T}_{C_k}} I(y_i \neq \hat{y}_i)/n_k$, n_k is the number of obs. in $\mathcal{T}_{C_k}$.
  - Compute $\text{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \text{Err}_k$.
- Example from ISL: Logistic regression with a non-linear (quadratic) decision boundary.

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

- For the simulated data, we can compute the *true* test error rate which is 0.201 and use the Bayes error rate, which is 0.133 as a benchmark. For real data, Bayes decision boundary and test error rates are unknown.
  Q: How to choose between the four logistic models?

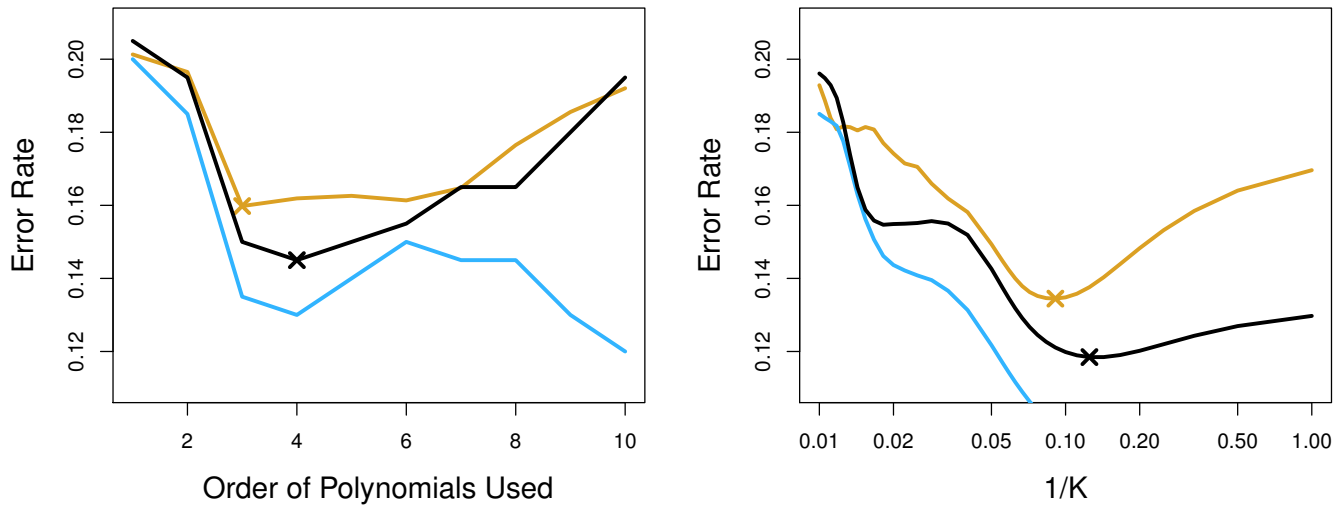
  A: use CV to make decision.

CV FOR CLASSIFICATION USING LOGISTIC REGRESSION



FIGUR: Simulated two-class bivariate data along with the logistic regression fit and Bayesian decision boundary (purple dashed line). Estimated classification boundaries from polynomial (degrees 1 to 4) logistic regression with the test errors estimated as $0.201, 0.197, 0.160$ and 0.162, respectively for each polynomial function. Bayes (irreducible) error rate is 0.133.

CV FOR CLASSIFICATION USING LOGISTIC REGRESSION (CONT.)



FIGUR: Test error (brown), training error (blue) and 10-fold CV error (black). Left: polynomial logistic regression based classifier. Right: KNN-classifier (K here is the number of neighbors, not a number of folds).

Right and wrong way to perform cross-validation

Example: Consider a two-class classification problem. Assume that the number of feature variables is $p = 5000$ and total sample size is $n = 50$, i.e. need to construct a classifier for a very high-dimensional case.

One possible strategy is to first select a subset of informative variables and then validate the performance of the resulting classifier, we call it for

Strategy I:

1. Screen the feature variables for their discriminative power. Select, say, 100 variables having largest correlation with the class label.
2. Construct a classifier based on the selected subset of highly discriminative features.
3. Perform cross-validation to estimate the unknown tuning parameters and to estimate Err of the final classifier.

Q: What is wrong with Strategy I ?

Answer: The discriminative feature variables were chosen after seeing all the data!

Correct way is **Strategy II:**

1. Split the samples into K folds randomly.
2. For each fold $k = 1, \ldots, K$
   - Find a subset of discriminative feature variables using all the samples except the kth.
   - Construct the classifier using all the samples except the kth. Apply the classifier to predict the class labels for the samples in kth fold.
   - Select the final model which minimizes CV error.

General approach to performing multi-step model fitting with CV:

- CV must be applied to the **entire sequence of model fitting steps**.
- Any supervised procedure which has seen the labels of the training data and used them, is a form of *training* and must be included in the validation procedure.