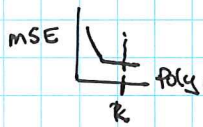
- Cross-Validation can be used for: - model selection (tuning parameter selection)
  - model assessment (estimating expected test MSE)

· It may be tempting to do both at once (ex. Fig 5.4 in slides)



· k degrees looks good (select) and then
· we (erroneously) use the MSE here for the estimated expected test MSE... → This is too optimistic / underestimates expected test MSE (too small)

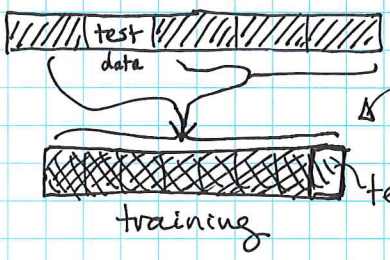
We should use a different data set to assess (ie a test set)

- The correct procedure: Double Cross Validation. (* "procedure" keyword)

- Use polynomial regression; we choose the degree by cross validation (Note that tuning / model selection is included in the procedure).

To assess the procedure, we need a second layer of cross (XV) validation

ex.



← Outer XV (5 fold) to assess procedure.
Inner XV (5 or 10 fold) for model selection.
ex. how many degrees is best? training different values of the tuning parameter to obtain the value that renders min. MSE

· Note that the selected model may be different for different training folds.
* If we include tuning param. in validation, then we need to do it in a separate (inner) XV loop.

BOOTSTRAP

- We have already seen that simulation is very useful for understanding the distribution of estimators - useful for inference, ex. for p-values, confidence intervals,

- In practice, we don't know the model that generated the data; We cannot do simulations (with that model).

- Efron's Nonparametric Bootstrap. (Efron 1979: simulating from an estimated model)
"Ordinary Bootstrap"

· Let data $Z_1, \dots, Z_n \overset{iid}{\sim} P$, which is a fixed, unknown distribution
ex. $Z_i = (X_i, Y_i)$, $X_i \in \mathbb{R}^p$, $Y_i \in \mathbb{R}$.

· Estimator or Statistical Procedure.

$\hat{\theta}_n = g(Z_1, \dots, Z_n)$, g is a known function. ex. $\hat{\theta}_n = \hat{\beta}_{n,LSQ} = (X^TX)^{-1}X^TY$.

$\hat{\theta}_n$ can be a vector or a function, ex. $\hat{\theta}_n = \hat{m}_n(\cdot)$ LOESS curve

Goal: get distribution of $\hat{\theta}_n$

- Efron Nonparametric Bootstrap, cont'd.

Solutions (for getting the distribution for $\hat{\theta}_n$)

① Exact distributions in certain special cases, (no bootstrap)
ex. $Z_1, \dots, Z_n \overset{iid}{\sim} N(\mu, \sigma^2)$ and $\hat{\theta}_n = \frac{1}{n}\sum_{i=1} Z_i = \bar{Z}_n$ (mean)

- (Linear combo of Gaussians is Gaussian) so $\hat{\theta}_n \sim N(\mu, \sigma^2/n)$ is exact. (estimate σ^2 from data $\mathbf{Z}$

$$\frac{\hat{\theta}_n - \mu}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1} \quad \text{exact.}$$

② Asymptotic approximations: $Z_1, \dots, Z_n \overset{iid}{\sim} P$, which is fixed, unknown.

(a) so $\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i = \bar{Z}_n$ estimator for μ. $E(Z_i) = \mu$, $Var(Z_i) = \sigma^2$

($\hat{\theta}_n$ is Gaussian because of the Central Limit Theorem s.t. $\hat{\theta}_n \overset{\approx}{\text{approx}} N(\mu, \sigma^2/n)$)

(b) $Z_1, \dots, Z_n \overset{iid}{\sim} P$, $\theta = \text{median}(P)$, f is density of P.

$\hat{\theta}_n$ = sample median of $Z_1, \dots, Z_n$

$\hat{\theta}_n \overset{\approx}{} N(\theta, \sigma^2_{approx}/n)$

$\sigma^2_{asymp} = \frac{E\left((Z-\theta)^2\right)}{4f(\theta)}$ ← awkward estimation.

③ If $\hat{\theta}_n$ is a "complicated" algorithm, then we may not even know the approximate distribution of $\hat{\theta}_n$.

⇒ Bootstrap (good for ② ③). Approach: simulate data from estimated model $\hat{P}$ (if we do not have P)

- Particularly in non-parametric bootstrap, we simulate from $\hat{P}_n$, which is the empirical distribution of $Z_1, \dots, Z_n$.
- $\hat{P}_n$ places mass $\frac{1}{n}$ at each data point. It is a discrete distribution on the datapoints
- Each point is equally likely (to be drawn).

- What does it mean to sample from $\hat{P}_n$?

It is uniform sampling with replacement from the data points. $\{Z_1, \dots, Z_n\}$. ex. if $\mathbf{Z} = \{Z_1, Z_2, Z_3\}$, we select for ex. $Z_1^* = Z_2$, $Z_2^* = Z_3$, $Z_3^* = Z_2$.
* means bootstrapped.

Efron's Non Parametric Bootstrap, cont'd

- **Bootstrap algorithm for estimator** $\hat{\theta}_n = g(\mathbb{Z})$

1. Generate a bootstrap sample $Z_1^*, \dots, Z_n^* \overset{iid}{\sim} \hat{P}_n$, where Z^* is a realization of Z_i in $\{Z_1, Z_2, \dots, Z_n\} := \mathbb{Z}$ (sampled with replacement from $\mathbb{Z}$)
2. Compute bootstrapped estimator $\hat{\theta}_n = g(Z_1^*, \dots, Z_n^*)$.

- Repeat 1,2 B times, where B can be large; this yields $\hat{\theta}_n^{*1}, \hat{\theta}_n^{*2}, \dots \hat{\theta}_n^{*B}$
- Bootstrap distribution P^* is the distribution of $\hat{\theta}_n$ induced by sampling from $\hat{P}_n$. P^* is a conditional distribution (given the data),

We can simulate P^* by using the bootstrap algorithm with a large B:

$$P(\sqrt{n}(\hat{\theta}_n - \theta) \le x) - P^*(\sqrt{n}(\hat{\theta}_n^* - \hat{\theta}_n) \le x) \xrightarrow{P} 0 \quad \text{as } n \to \infty$$

true rescaled distribution (centered around θ) ← true distr.

bootstrap rescaled distribution (centered around $\hat{\theta}_n$)

except for the center, they are the same

Now we can look at Var of $P^*(\sqrt{n}(\hat{\theta}_n^* - \hat{\theta}_n) \le x)$ distribution, bias in P^*

R code: $\mathrm{Var}(\alpha X + (1-\alpha)Y)$ ← want this to be minimal,

The minimum is attained at some value

$$\alpha = \frac{\sigma_y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}}$$

, which we compute and assess

(replace = false for CV)