Statistical Machine Learning

Lecture 6: Model Assessment and Selection

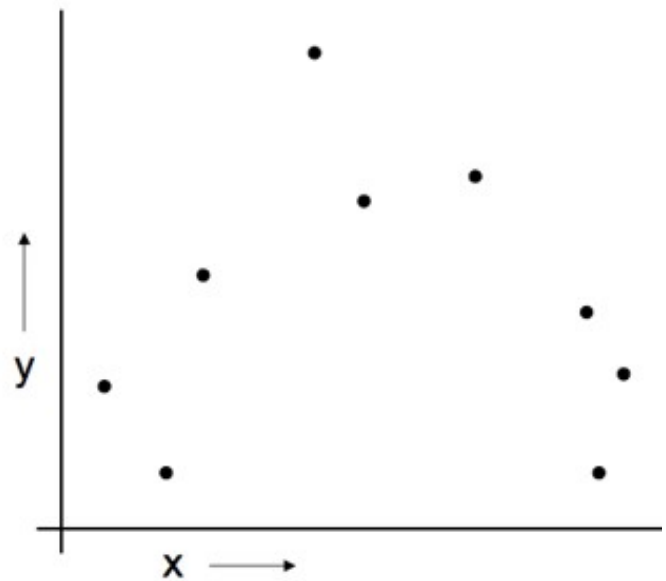
W.Q.Cui Research Group

Department of Statistics and Finance
University of Science and Technology of China

2018 Autumn

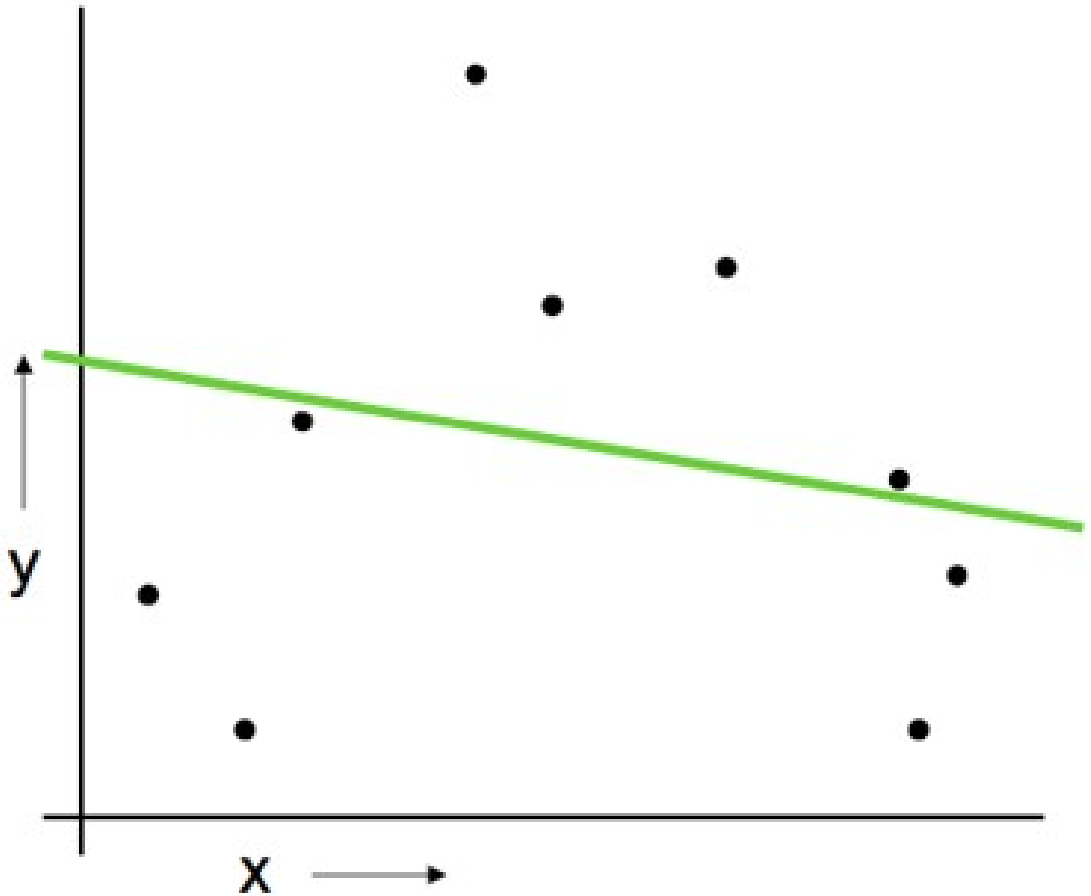
Contents

A Regression Problem

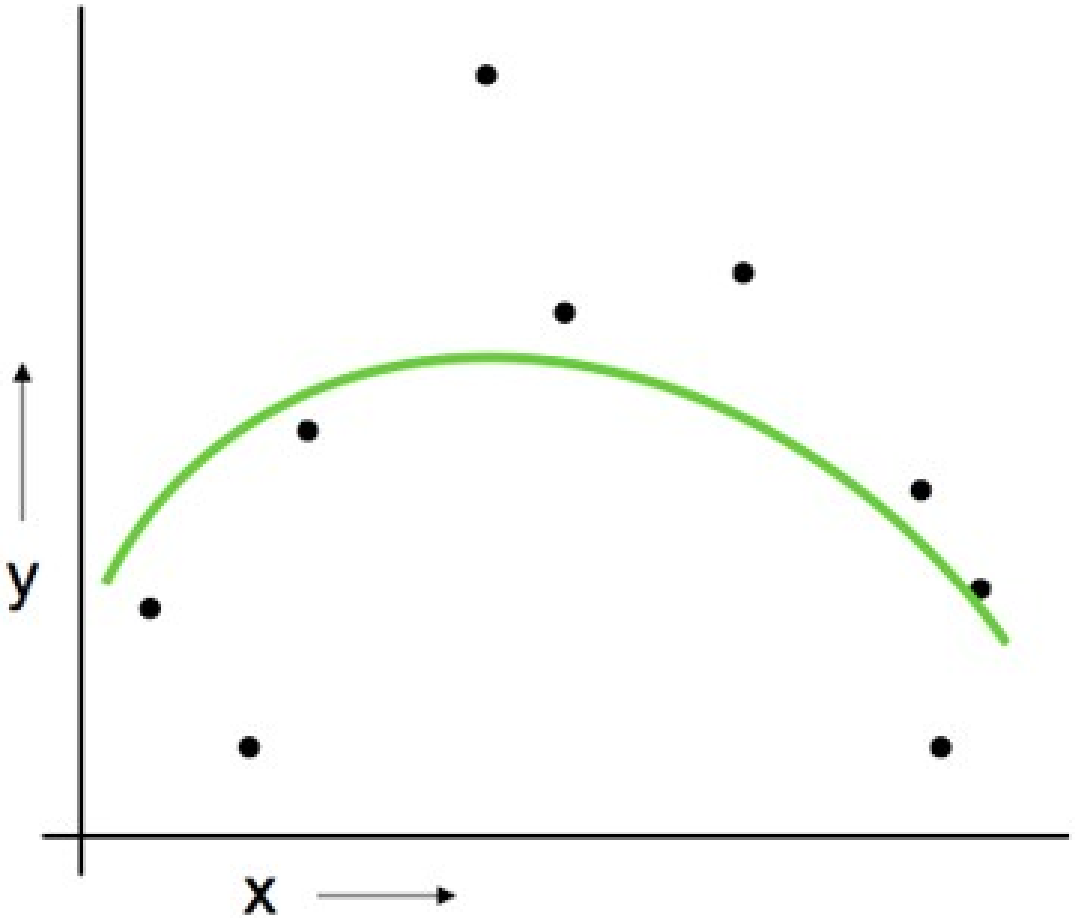


- $y = f(x) + \epsilon$
- Can we learn f from this data?
- Let's consider three methods

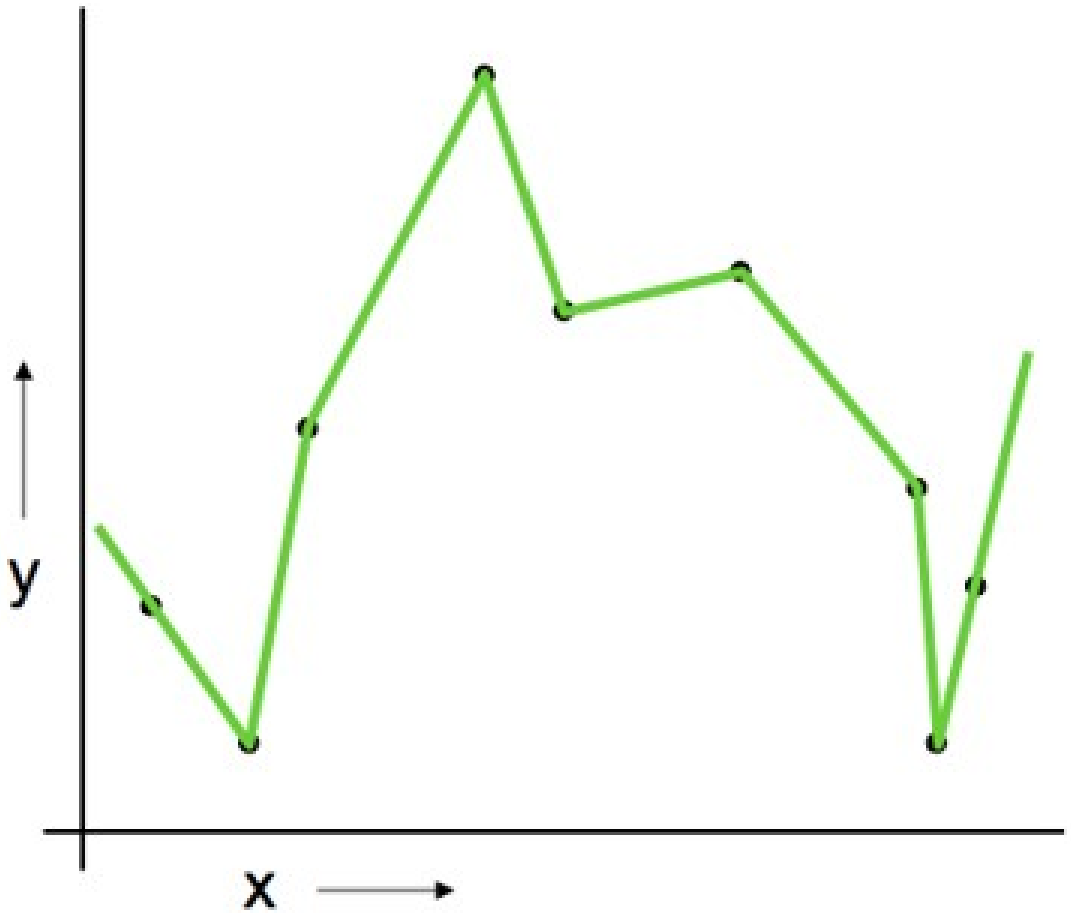
Linear Regression



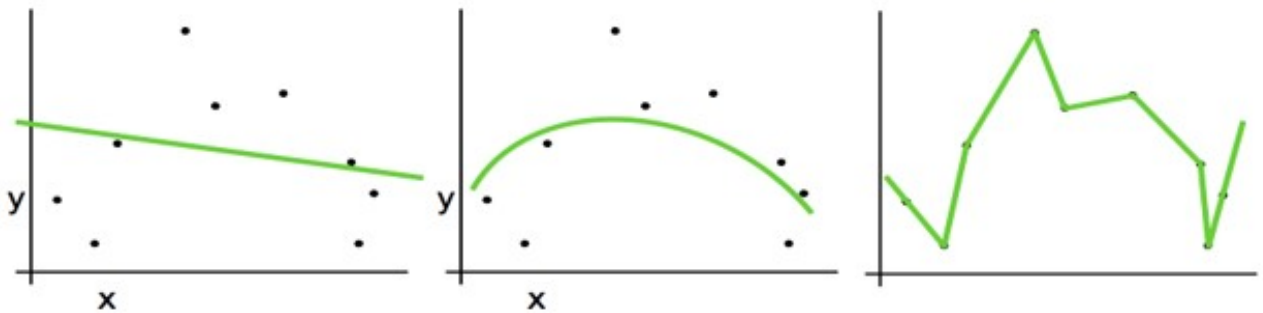
Quadratic Regression



Joining the dots



Which is best?



Why not choose the method with the best fit to the data?

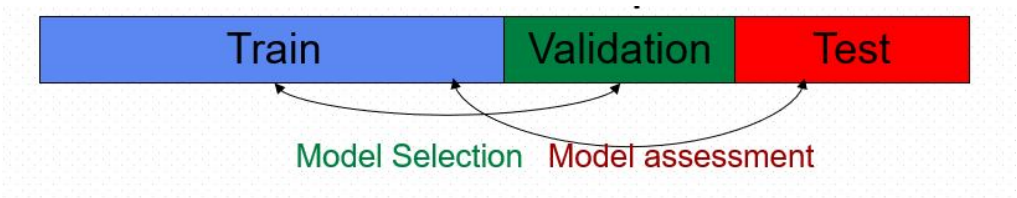
How well are you going to predict future data drawn from the same distribution?

Model Selection and Assessment

- Model Selection: Estimating performances of different models to choose the best one(produces the minimum of the test error)
- Model Assessment: Having chosen a model, estimating the predictive error on new data

Why Errors

- Why do we want to study errors?
- In a data-rich situation split the data:



- But, that's not usually the case

Remainder of the chapter:Data-poor situation $\Rightarrow$ Approximation of validation step either analytically(AIC, BIC, MDL, SRM) or by efficient sample reuse(cross-validation, bootstrap)

Overall Motivation

- Errors
  - Measurement of errors(Loss Functions)
  - Decomposing Test Error into Bias & Variance
- Estimating the true error
  - Esimating in-sample error(analytically)
    AIC, BIC, MDL, SRM with VC
  - Estimating extra-sample error(efficient sample reuse)
    Cross Validation & Bootstrap

Measuring Errors: Loss Function

Typical regression loss functions

- Squared error:
$$L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$$
- Absolute error:
$$L(Y, \hat{f}(X)) = |Y - \hat{f}(X)|$$

Measuring Errors: Loss Function

Typical classification loss functions

- 0-1 Loss:

$$L(G, \hat{G}(X)) = I(G \neq \hat{G}(X))$$

- Log-Likelihood(cross-entropy loss/deviance):

$$L(G, \hat{G}(X)) = -2log\hat{p}_G(X)$$

The Goal: Low Test Error

- We want to minimize general error or test error:

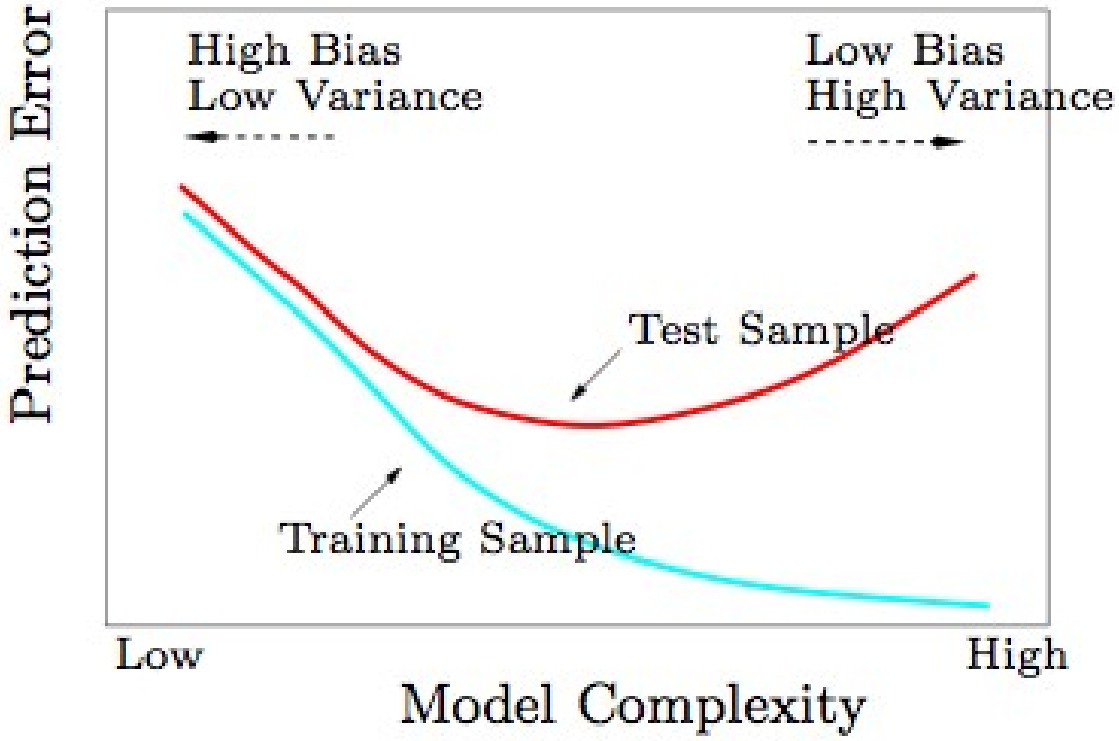
$$err = E(L(Y, \hat{f}(X)))$$

- But all we really know is the training error:

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

- **And this is a bad estimation of test error.**

Bias, Variance & Complexity



Training error can always be reduced when increasing model complexity, but risks over-fitting.
Typically, $\overline{err} < err$

Decomposing Test Error

Model:

$$Y = f(X) + \epsilon, E(\epsilon) = 0, Var(\epsilon) = \sigma_\epsilon^2$$

For squared-error loss and additive noise:

$$\begin{aligned} err(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\ &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\ &= \sigma_\epsilon^2 + Bias^2(\hat{f}(x_0)) + Var(\hat{f}(x_0)) \end{aligned}$$

- σ_ϵ^2: Irreducible error of target Y
- $Bias^2(\hat{f}(x_0))$: Deviation of the average estimate from the true function's mean
- $Var(\hat{f}(x_0))$: Expected squared deviation of our estimate around its mean

Further Bias Decomposition

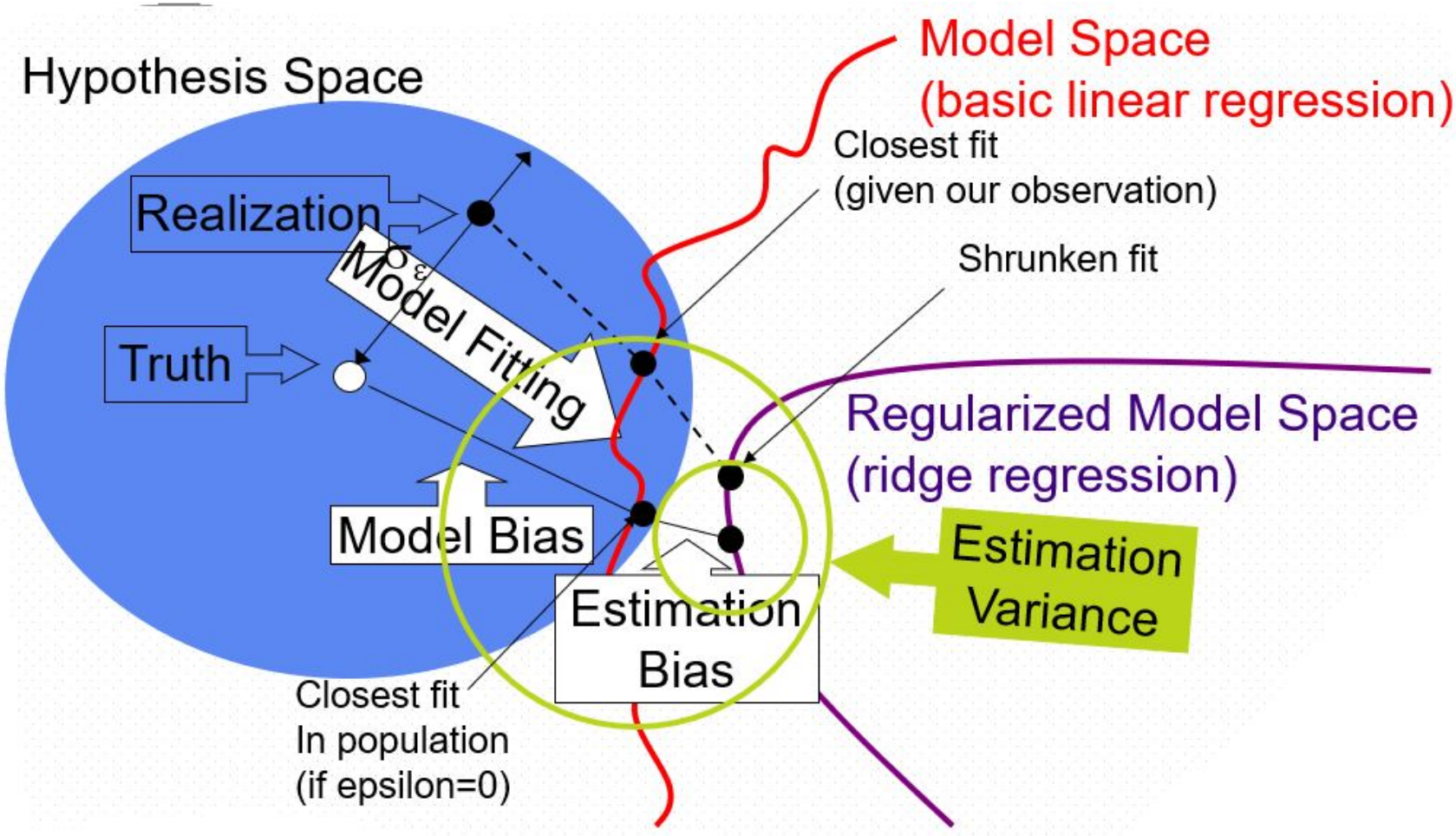
For linear models(eg. Ridge), bias can be further decomposed:

$$\begin{aligned} Bias^2 &= E_{x_0}[f(x_0) - E\hat{f}_\alpha(x_0)]^2 \\ &= E_{x_0}[f(x_0) - \beta_*^T x_0]^2 + E_{x_0}[\beta_*^T x_0 - E\hat{\beta}_\alpha^T x_0]^2 \end{aligned}$$

- $E_{x_0}[f(x_0) - \beta_*^T x_0]^2$: Average Model Bias
- $E_{x_0}[\beta_*^T x_0 - E\hat{\beta}_\alpha^T x_0]^2$: Average Estimation Bias. For standard linear regression, Estimation Bias $= 0$.
- β_* is the best fitting linear approximation

$$\beta_* = argminE(f(X - \beta^T X)^2$$

Graphical Representation of Bias & Variance



Bias & Variance Decomposition Examples

- kNN Regression

$$err(x_0) = \sigma_\epsilon^2 + [f(x_0) - \frac{1}{k}\sum_{\ell=1}^{k} f(x_{(\ell)})]^2 + \frac{\sigma_\epsilon^2}{k}$$

- Linear Regression

$$err(x_0) = \sigma_\epsilon^2 + [f(x_0) - E\hat{f}_p(x_0)]^2 + ||h(x_0)||^2\sigma_\epsilon^2$$

Average error over the training set:

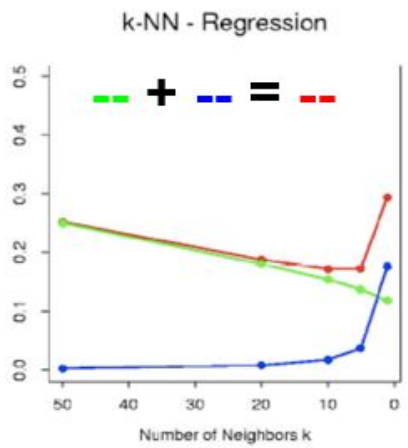
$$\frac{1}{N}\sum_i err(x_i) = \sigma_\epsilon^2 + \frac{1}{N}\sum_i [f(x_i) - E\hat{f}(x_i)]^2 + \frac{p}{N}\sigma_\epsilon^2$$

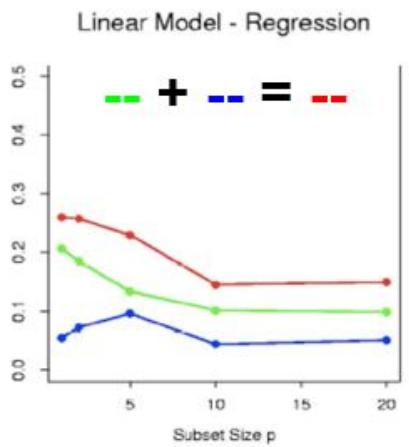
$$\hat{f}_p(x_0) = x_0^T(X^TX)^{(-1)}X^Ty = h(x_0)y$$

$h(x_0) = x_0^T(X^TX)^{(-1)}X^T$ are the linear weights on y.

Simulated Example of Bias Variance Decomposition

Regression with squared error loss

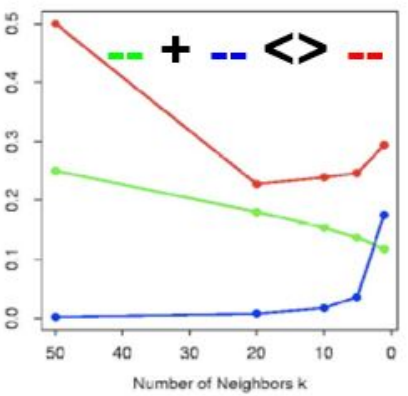


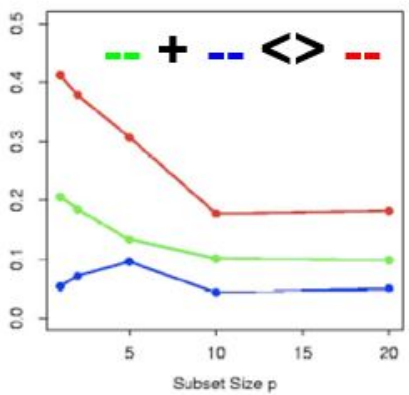


Prediction error
Bias^2
Variance

Classification with 0-1 loss

Estimation errors on the right side of the boundary don't hurt!





Bias-Variance different for *0-1 loss* than for *squared error loss*

Optimism of The Training Error Rate

Typically: training error $<$ true error (same data is being used to fit method and assess its error)

$$\overline{err} < err$$

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

$$err = E[L(Y, \hat{f}(X))]$$

Estimating Test Error

Can we estimate the discrepancy between $\overline{err}$ and err(extra-sample error) err_{in} — in-sample error:

$$err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_y E_Y^{new} L(Y_i^{new}, \hat{f}(x_i))$$

$$= E_y(\overline{err}) - \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i)$$

Optimism

Summary:

$$err_{in} = E_y(\overline{err}) + \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i)$$

For squared error, 0-1 and other loss functions:

$$optimism : op \equiv err_{in} - E_y(\overline{err})$$

$$\Rightarrow op = \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i)$$

For linear fit with d independent inputs/bias functions:

$$err_{in} = E_y(\overline{err}) + \frac{2}{N} d\sigma_\epsilon^2$$

- Optimism $\uparrow$ linear with $\#d$
- Optimism $\downarrow$ as training sample size $\uparrow$

Ways to Estimate Prediction Error

- In-sample error estimates:
  - AIC
  - BIC
  - MDL
  - SRM
- Extra-sample error estimates:
  - Cross-Validation
    - leave-one-out
    - k-fold
  - Bootstrap

Estimates of In-Sample Prediction Error

- General form of the in-sample estimate:

$$\hat{Err}_{in} = \overline{err} + \hat{op}$$

- For linear fit(C_p Statistic):

$$C_p = \overline{err} + \frac{2d}{N}\hat{\sigma}_\epsilon^2$$

AIC & BIC

- Akaike Information Criterion (AIC)

$$AIC = -\frac{2}{N} \cdot ln(likelihood) + 2 \cdot \frac{d}{N}$$

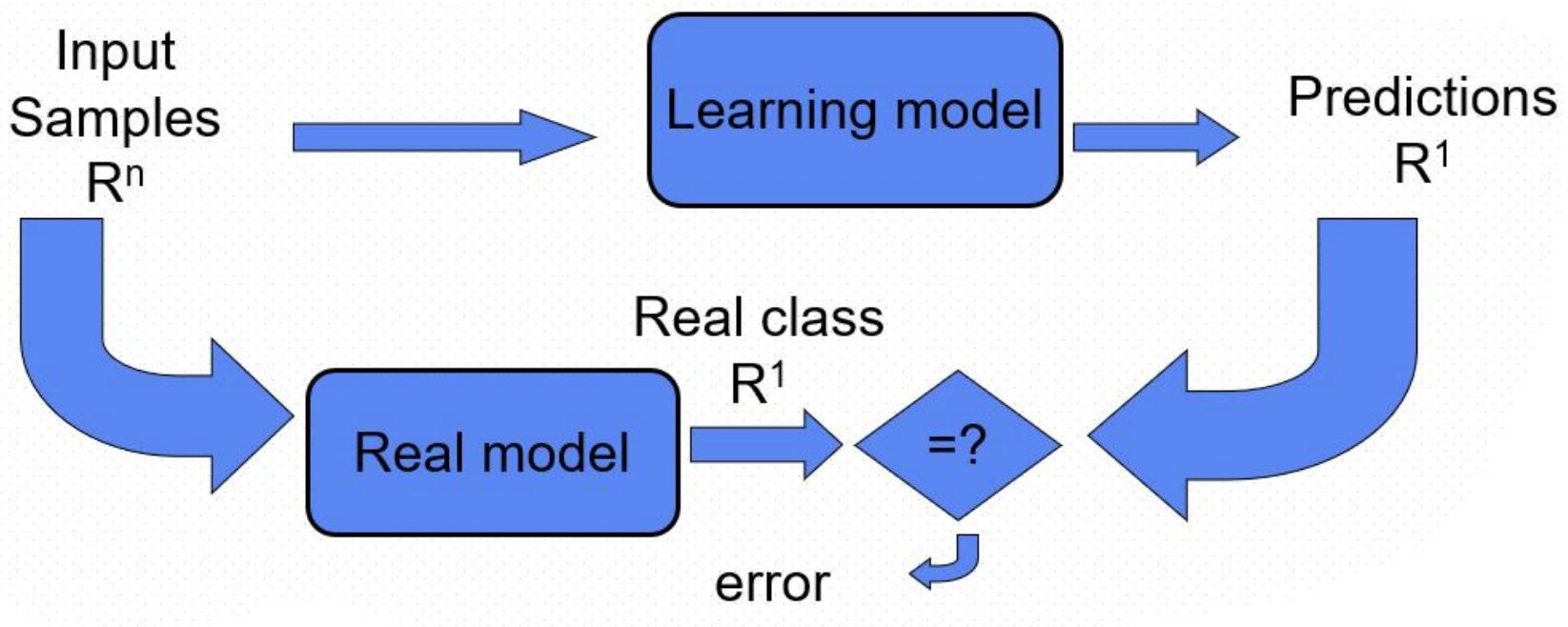
- Bayesian Information Criterion (BIC)

$$BIC = -2 \cdot ln(likelihood) + ln(N)d$$

where d is the number of the parameter, N is the sample size.

MDL (Minimum Description Length)

- Regularity $\sim$ Compressibility
- Learning $\sim$ Finding regularities



MDL (Minimum Description Length)

- Regularity $\sim$ Compressibility
- Learning $\sim$ Finding regularities

$$length = -lnPr(y|\theta, M, X) - lnPr(\theta|M)$$

Length of transmitting the discrepancy given the model + optimal coding under the given model

Description of the model under optimal coding

MDL principle: choose the model with the minimum description length

Equivalent to maximizing the posterior: $Pr(y|\theta, M, X)Pr(\theta|M)$

SRM with VC (Vapnik-Chernovenkis) Dimension

- Vapnik showed that with probability $1 - \eta$

$$err_{true} \leq err_{train} + \frac{\epsilon}{2}(1 + \sqrt{1 + \frac{4 \cdot err_{train}}{\epsilon}}) = I + II$$

where

$$\epsilon = a_1 \frac{h[ln(a_2 N/h) + 1] - ln(\eta/4)}{N}$$

and II is h-VC dimension (measure of $f's$ power). As h increases, $I \downarrow$ and $II \uparrow$.

- A method of selecting a class F from a family of nested classes

err_{in} Estimation

A trade-off between the fit to the data and the model complexity

$$AIC = \overline{err} + 2 \cdot \frac{d}{N} \cdot \hat{\sigma}_e$$

$$BIC = -2 \cdot ln(likelihood) + ln(N)d$$

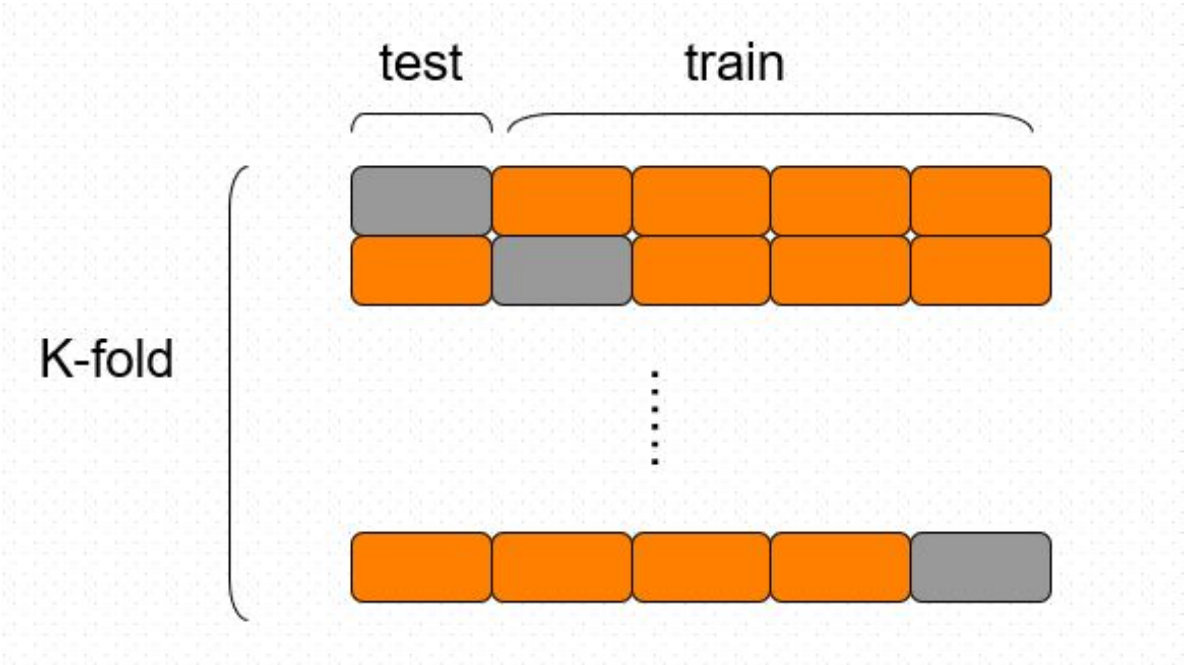
$$MDL\ length = -lnPr(y|\theta, M, X) - lnPr(\theta|M)$$

$$VC : err_{true} \leq err_{train} + \frac{\epsilon}{2}(1 + \sqrt{1 + \frac{4 \cdot err_{train}}{\epsilon}})$$

Estimation of Extra-Sample Err

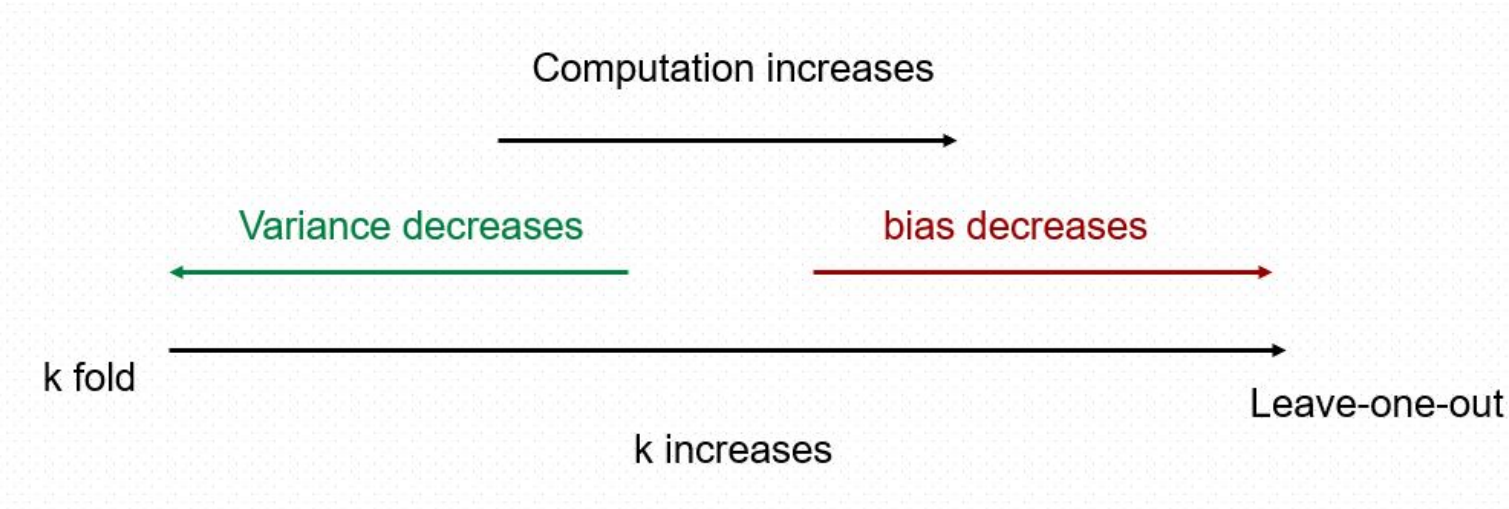
- Cross Validation
- Bootstrap

Cross Validation

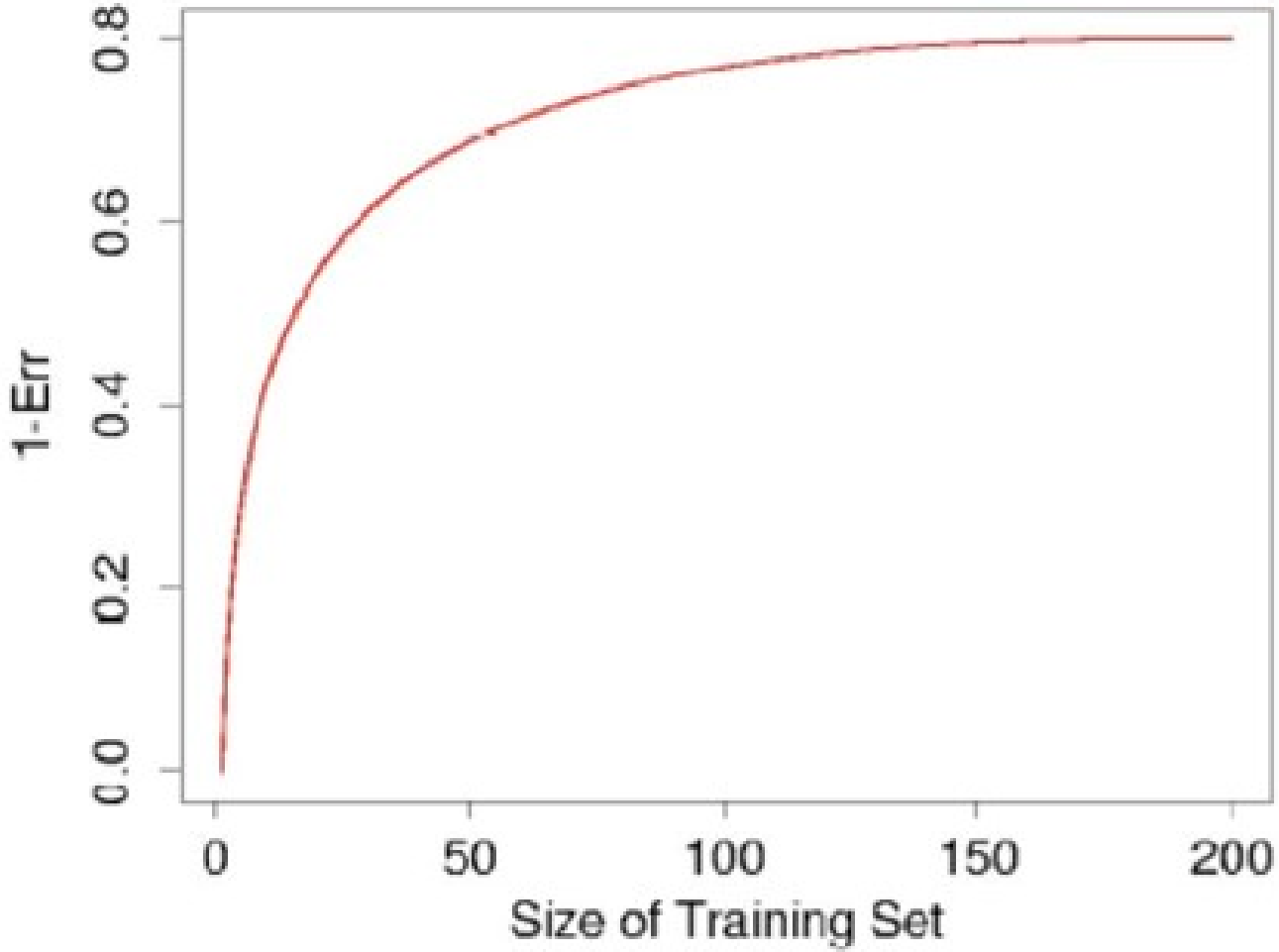


$$CV(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha))$$

How Many Folds?



Cross-Validation: Choosing K



Popular choices for $K : 5, 10, N$

Generalized Cross-Validation

- LOO-CV(Leave one out Cross Validation) can be computational expensive for linear fitting with large N
- Linear fitting $\hat{y} = Sy$, where S is a smoother matrix.
- For linear fitting under squared-error loss:

$$\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}^{-i}(x_i)]^2 = \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}]^2$$

$S_{ii} = i$th diagonal element of S

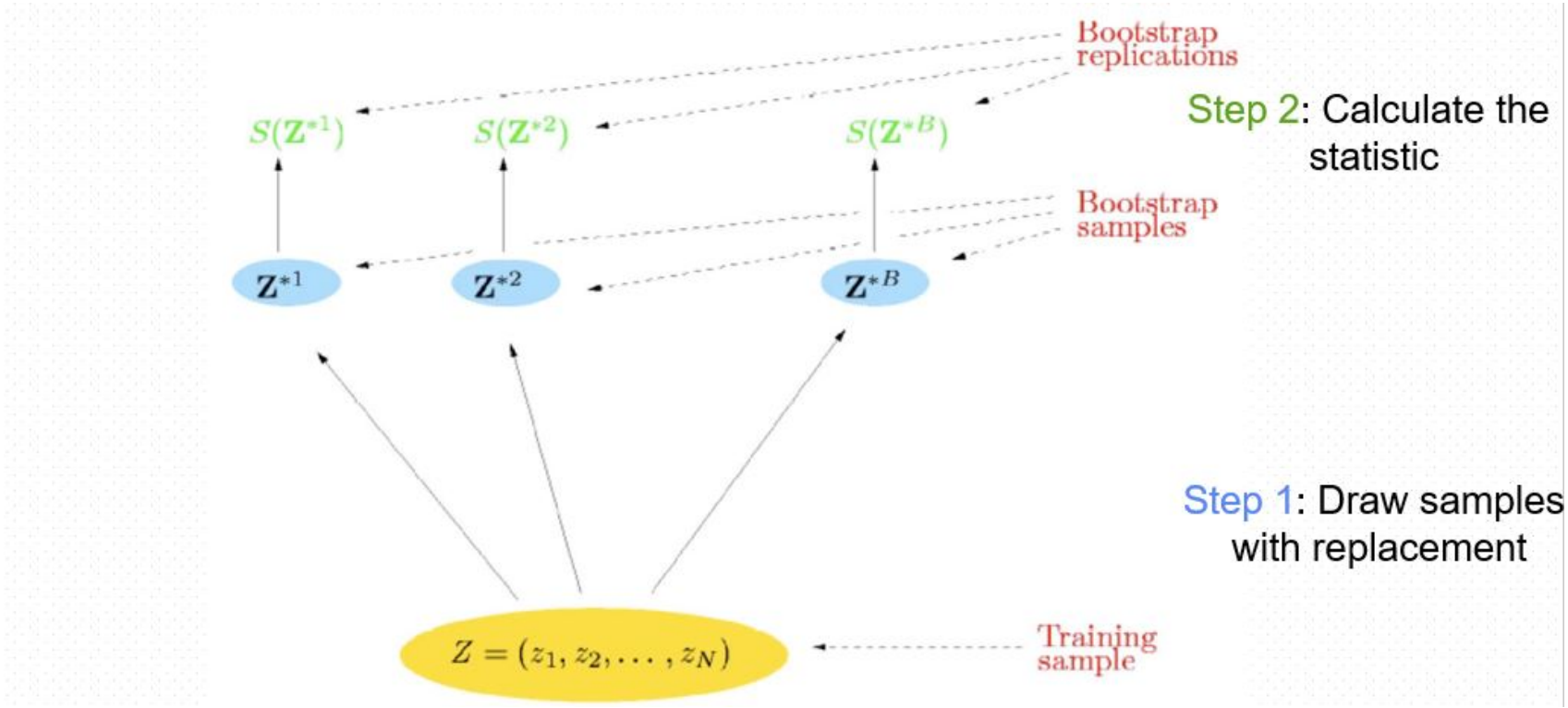
$$\lambda = \frac{tr(S)}{N}$$

- GCV provides a computationally cheaper approximation

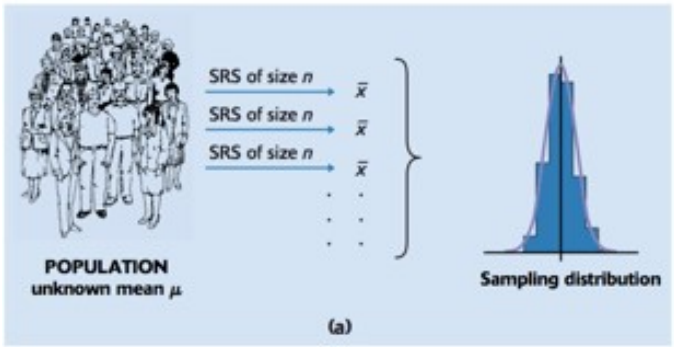
$$GCV = \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - \lambda}]^2$$

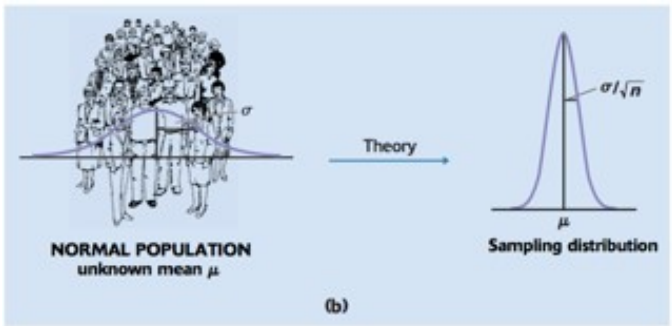
Bootstrap: Main Concept

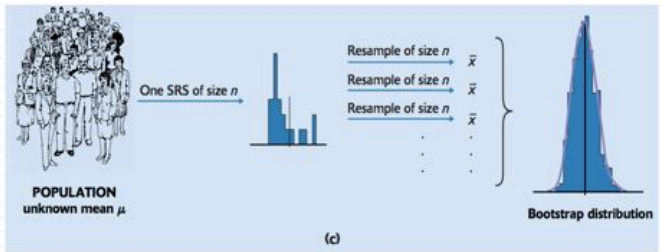
"The bootstrap is a computer-based method of statistical inference that can answer many real statistical questions without formulas"-(An Introduction to the Bootstrap, Efron and Tibshirani, 1993)



How is it coming







- (a) $\rightarrow$ Sampling distribution of sample mean $\bar{x}$, but in practice we cannot afford large number of random samples
- (b) $\rightarrow$ The theory tells us the sampling distribution
- (c) $\rightarrow$ The sample stands for the population and distribution of $\bar{x}$ in many resamples stands for the sampling stands for the sampling distribution

Bootstrap: Error Estimation with err_{boot}

$$\widehat{Var}[S(Z)] = \frac{1}{B-1}\sum_{b=1}^{B}(S(Z^{*b}) - \bar{S}^*)$$

where $\bar{S}^* = \frac{1}{B}\sum_{b=1}^{B} S(Z^{*b})$, $Var_{\hat{F}}(S(Z))$ depends on the unknow true distribution F.

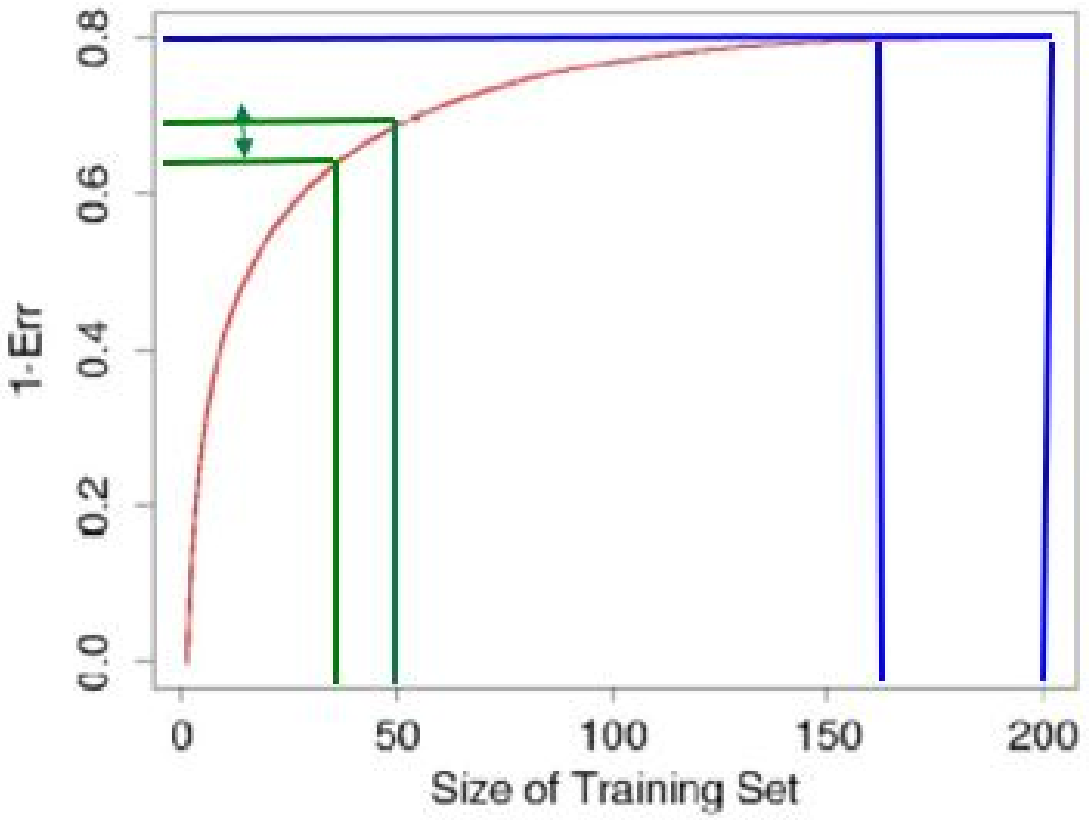
A straightforward application of bootstrap to error prediction

$$\widehat{err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i))$$

Bootstrap: Error Estimation with $Err^{(1)}$

A CV-inspired improvement on err_{boot}:

$$\widehat{err}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{f}^{*b}(x_i))$$



Bootstrap: Error Estimation with $Err^{(.632)}$

An improvement on $err^{(1)}$ in light-fitting cases:

$$\widehat{err}^{(.632)} = 0.368 \cdot \overline{err} + 0.632 \cdot \widehat{err}^{(1)}$$

- N = size of data points $Z = (z_1, \cdots, z_n)$
- Probability of z_i NOT being chosen when 1 point is uniformly sampled from Z : $(1 - \frac{1}{N})$
- Probability of z_i NOT being chosen when Z is sampled N times : $(1 - \frac{1}{N})^N$
- Probability of z_i NOT being chosen AT LEAST once when Z is sampled N times : $1 - (1 - \frac{1}{N})^N$
  so,

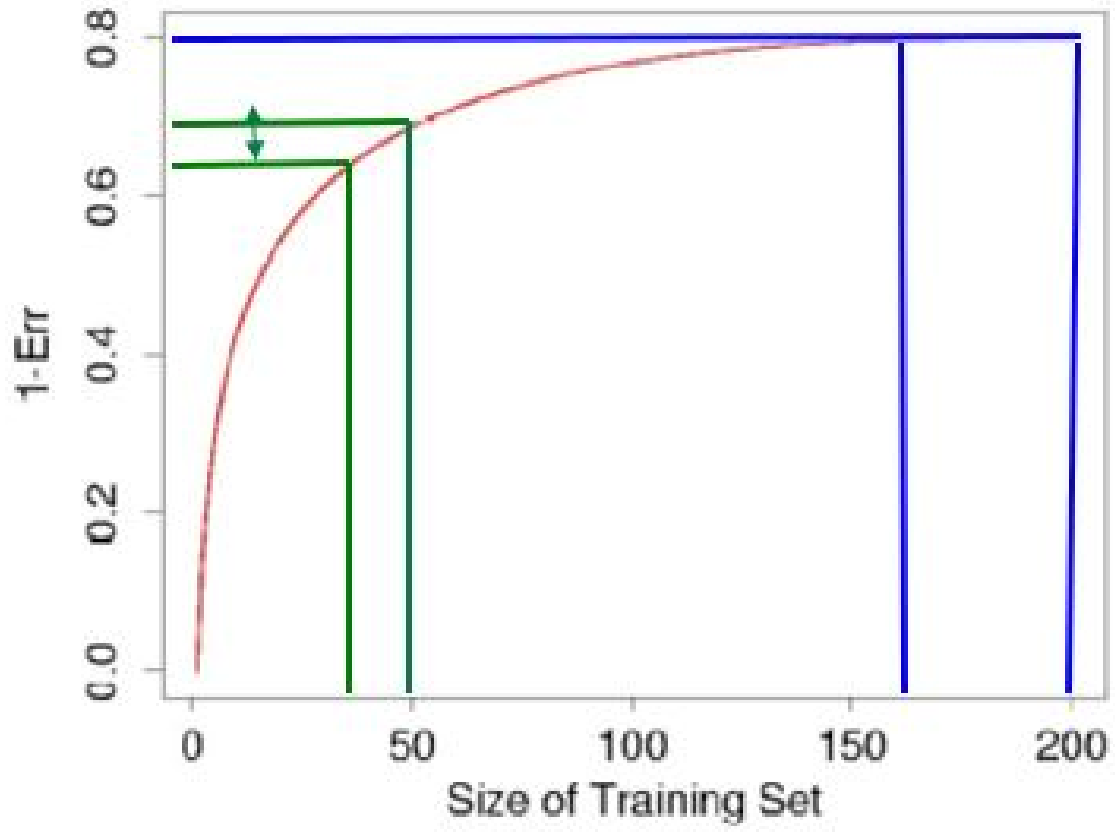
$$\begin{aligned}\widehat{err}^{(.632)} &= \overline{err} + (1 - e^{-1}) \cdot (\widehat{err}^{(1)} - \overline{err}) \\ &= 0.368 \cdot \overline{err} + 0.632 \cdot \widehat{err}^{(1)}\end{aligned}$$

Bootstrap: Error Estimation with $err^{(.632+)}$

An improvement on $err^{(.632)}$ by adaptively accounting for overfitting

- Depending on the amount of overfitting, the best error estimate is as little as $err^{(.632)}$, or as much as $err^{(1)}$, or something in between
- $err^{(.632+)}$ is like $err^{(.632)}$ with adaptive weights, with $err^{(1)}$ weighted at least 0.632
- $err^{(.632+)}$ adaptively mixes training error and leave-one-out error using the relative overfitting rate(R)

Bootstrap: Error Estimation with $err^{(.632+)}$



$err^{(.632+)}$ ranges from $err^{(.632)}$ if there is minimal overfitting($R = 0$), to $err^{(1)}$ if there is maximal overfitting($R = 1$).

Cross Validation & Bootstrap

Why bother with cross-validation and bootstrap when analytical estimates are known?

- AIC, BIC, MDL, SRM all requires knowledge of d, which is difficult to attain in most situations.
- Bootstrap and cross validation gives similar results to above but also applicable in more complex situation.
- Estimating the noise variance requires a roughly working model, cross validation and bootstrap will work well even if the model is far from correct.

Conclusion

- Test error plays crucial roles in model selection
- AIC, BIC and SRMVC have the advantage that you only need the training error
- If VC-dimension is known, then SRM is a good method for model selection −− requires much less computation than CV and bootstrap, **but is wildly conservative**
- Methods like CV, Bootstrap give tighter error bounds, **but might have more variance**
- Asymptotically AIC and Leave-one-out CV should be the same
- Asymptotically BIC and a carefully chosen k-fold should be the same
- BIC is what you want if you want the best structure instead of the best predictor
- Bootstrap has much wider applicability than just estimating prediction error