Regression and Prediction ML

Version 1.0

Dr. Ye Luo
HKU Business School

Jan 2023

Statistical Model and Prediction

- A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data (and similar data from a larger population). A statistical model represents, often in considerably idealized form, the "data-generating process", or DGP in short.
- A statistical model is usually specified as a mathematical relationship between one or more random variables and other non-random variables.
- The goal is to establish a map $f : X \mapsto Y$ where X, Y could be multi-dimensional object, with f being referred as the prediction function.

Goal and Specifications

- The goal of statistical machine learning is to form a process that learns the function $f : X \mapsto Y$.
- The learning process is an algorithm that attempts to solve a minimization, maximization problem, or an equation.
- The minimization and maximization problem can also be viewed as solving an equation in the gradient space of the objective function to be minimized or maximized.

The Loss Function

- In mathematical optimization and decision theory, a loss function or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.
- The loss function measures how well a model $f(X)$ explains about the target variable Y. Namely, $L(f(X), Y)$ maps the pair (X, Y) to a real number given the function $f(\cdot)$.
- The loss function can be evaluated in the sample $(X_1, Y_1), ..., (X_n, Y_n)$. That is to say,
$$L_n(f) := \frac{1}{n} \sum_{i=1}^{n} L(f(X_i), Y_i).$$
- Machine Learning attempts to minimize $L_n(f)$ over a space $\mathcal{F}$ of possible functions of f.

$$\hat{f} = argmin_{f \in \mathcal{F}} L_n(f).$$

Leading Example: Regression

- One specific case of the loss function $L(\cdot, \cdot)$ can be specified as:

$$L(f(X), Y) = (Y - f(X))^2,$$

  often referred as the squared loss or L-2 loss.
- Such loss function is popular, as the squared loss favors the prediction $f(X)$ being close to Y rather much more than being away from Y.
- So the $\hat{f}$ that solves

$$\hat{f} = argmin_{f \in \mathcal{F}} L_n(f) = argmin_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2.$$

- Solving $\hat{f}$ described in the above equation is called a "regression" process.

Interpreting Regression as Conditional Mean Estimation

- Suppose $X, Y \sim \pi(X, Y)$ as a fixed distribution of data. $\pi(\cdot, \cdot)$ is a probability distribution function of (X, Y).
- Consider the following problem: $\hat{f} := argmin_f \mathbb{E}_{X,Y\sim\pi(\cdot,\cdot)}[(Y - f(X))^2]$.
- The problem can be boiled down by iterated expectations:

$$\begin{aligned}\hat{f} &:= argmin_f \mathbb{E}_{X,Y\sim\pi(\cdot,\cdot)}[(Y - f(X))^2] \\ &= argmin_f \mathbb{E}_{x\sim\pi_X(\cdot)}[\mathbb{E}_{Y\sim\pi_{Y|X}(\cdot)}[(Y - f(X))^2|X]].\end{aligned}$$

Regression as Conditional Mean Estimation

- $f(X)$ is a constant conditional on the observation of X. What is the scalar a that solves:

$$\mathbb{E}_{Y \sim \pi_{Y|X}(\cdot)}[(Y-a)^2].$$

- Take derivative with respect to a, from the equation in the above, we have that:

$$a = \mathbb{E}_{Y \sim \pi_{Y|X}(\cdot)}\mathbb{E}[Y] = \mathbb{E}[Y|X].$$

- Therefore, the best function f is the one such that $f(X) = \mathbb{E}[Y|X]$ for all X.
- That said, regression (loss function as squared loss) is to solve for the mean of Y conditional on X.

Linear regression: a special form of regression

- In general, X could be multi-dimensional. Therefore, estimating the conditional mean $\mathbb{E}[Y|X]$ can be challenging.
- In practice, a functional form assumption needs to be taken. The most simple form one can have is to assume a linear functional form:

  $$\mathbb{E}[Y|X] = X^\intercal \beta,$$

  for some parameter $\beta \in \mathbb{R}^p$ given $dim(X) = p$.
- Typically

  $$\mathbb{E}[Y|X]$$

  is non-linear and non-parametric. Therefore, the linear functional form $X^\intercal \beta$ could be mis-specified. However, linear regression still produces the "best" linear predictor within the class of linear functions.

A Review of Linear Regression

- Assume that $Y_i = X_i\beta + \epsilon_i,\ i = 1, 2, ..., n$.
- The residual ϵ_i is assumed to be independent and identically distributed (i.i.d.), with $\mathbb{E}[\epsilon_i|X_i] = 0$.
- The linear regression solves:

$$\hat{\beta} = argmin_{\beta} \frac{1}{n} \sum_{i=1}^{n} (Y_i - X_i^{\intercal}\beta)^2.$$

Algebraic Form

- $X_i = \begin{pmatrix} X_{i1} \\ X_{i2} \\ ... \\ X_{ip} \end{pmatrix}$ is a $p \times 1$ vector, $i = 1, 2, ..., n$.
- Define the matrix $X = \begin{pmatrix} X_{11} & X_{12} & ... & X_{1p} \\ ... & ... & ... & ... \\ X_{n1} & X_{n2} & ... & X_{np} \end{pmatrix}$ as a $n \times p$ matrix, and

  $Y = \begin{pmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{pmatrix}$ as a $n \times 1$ matrix. One can rewrite the optimization problem into an algebraic form:

$$\hat{\beta} = argmin_\beta (Y - X\beta)^\intercal (Y - X\beta).$$

Algebraic Solution to Linear Regression

- Take derivative with respect to β, we have that:

$$-2X^\intercal(Y - X\beta) = 0$$

- We have that

$$\hat{\beta} = (X^\intercal X)^{-1} X^\intercal Y.$$

- This formula works for all standard linear regression problems. The algebraic computation involves matrix multiplication and matrix inversion. Computing $(X^\intercal X)^{-1}$ could be complex and difficult in certain scenarios.
- What happens if two columns of X are identical? $X^\intercal X$ is singular, or non-invertible, leading to non-unique solution of linear regression.

Correct Specification

- If the specification is correct, i.e.,

$$Y_i = X_i\beta + \epsilon_i.$$

- Then, plug in this into the formula of $\hat{\beta}$, we have that:

$$\hat{\beta} = (X^\intercal X)^{-1}X^\intercal Y = \beta + (X^\intercal X)^{-1}X\epsilon,$$

with $\epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \dots \\ \epsilon_n \end{pmatrix}$ as a $n \times 1$ matrix.

- $\mathbb{E}[\epsilon|X] = 0$ implies that $\mathbb{E}[(X^\intercal X)^{-1}X\epsilon] = 0$. That is to say, $\mathbb{E}[\hat{\beta}] = \beta$: unbiasness of linear regression.

Convergence Speed and Inference

- The estimation error between $\hat{\beta}$ and β is:

$$(X^\intercal X)^{-1} X\epsilon = (\frac{1}{n} X^\intercal X)^{-1} \frac{1}{n} X\epsilon.$$

- $\frac{1}{n} X^\intercal X = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^\intercal$ converges to a $p \times p$ matrix $\Omega := \mathbb{E}[X_i X_i^\intercal]$ by Law of Large Numbers (LLN).
- $\frac{1}{\sqrt{n}} X\epsilon = \sqrt{n}(\frac{1}{n} \sum_{i=1}^{n} X_i \epsilon_i) \rightsquigarrow N(0, \Omega)$ by Central Limit Theorem.
- Therefore, we have:

$$\sqrt{n}(X^\intercal X)^{-1} X\epsilon \rightsquigarrow \Omega^{-1} N(0, \Omega) = N(0, \Omega^{-1}).$$

Simple Extension: Series Regression

- $Y = X\beta + \epsilon$ assumes linear functional form of $\mathbb{E}[Y|X]$. What if researchers want to consider non-linearity?
- For one dimensional x, any analytical function can be written as “infinite” linear combinations of series of x. For example, polynomials: $x, x^2, x^3, ...$
- We can use update to K polynomial terms to approximate for $\mathbb{E}[Y|X]$.

Mathematical Foundations

- Taylor expansion:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + ...$$

- The large K is, the smaller the approximation error is. Flexibility of functional form increases ability to approximate the "truth".
- Series regression:

$$min_{\beta_0,\beta_1,...,\beta_K} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i - \beta_2 x_i^2 - ... - \beta_K x_i^K)^2.$$

Multi-dimensional case - General Series

- For general $x = (x_1, ..., x_p)$, a similar approach can be applied.
- Standard series regression: second order terms -

$$x_1 x_2, ..., x_1 x_p, x_2 x_3, ..., x_2 x_p, ..., x_{p-1}, x_p.$$

  There are $p(p-1)/2$ terms.
- How about K^{th} order terms: We have

$$C_p^K := \frac{p(p-1)...(p-K+1)}{K(K-1)...1}$$

  terms.
- The number of terms needed increases quickly of p, K - This is referred as "Curse of Dimensionality": the larger the dimensionality is (p here), the more difficult to estimate the target of interest.

Linear versus Non-Linear

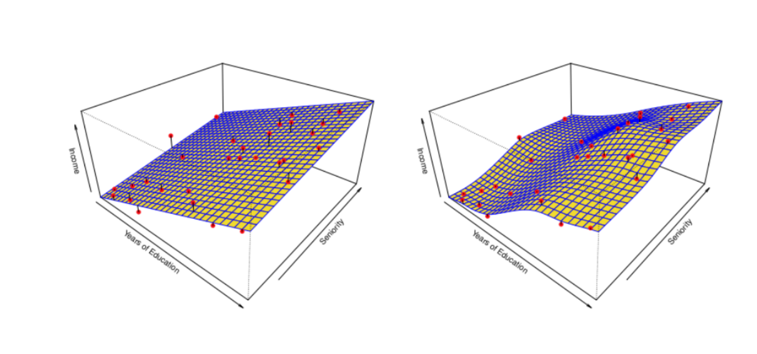


Figure 1: Linear versus non-linear regressions

Taming Dimensionality

- There are a few ways to tame Curse of dimensionality, which has been studied in the past 30 years in the related fields.
- In Series regression, one simple way is to consider "non-linear additive model":

$$y_i = \beta_0 + \sum_{j=1}^{p} (x_j, ..., x_j^K)^\intercal \beta_j,$$

  with $\beta_j \in \mathbb{R}^K$.
- The dimension is $pK + 1$, which is linear in p, K, rather than polynomial.

Other Popular Series

- Splines: $x \cdot 1(x > t_j)$ for some fixed t_j. A evenly spaced Spline: $t_j, j = 1, 2, ..., p$ obeys that $t_{j+1} = t_j + d$ for fixed distance d.
- Fourier Series: $cos(jx), sin(jx)$, $j = 1, 2, ..., p$.
- Cubic splines: $x^3 \cdot 1(x > t_j)$.

Basic Terminologies and Key Statistics

- Residual Sum of Squares (RSS) is defined as:

$$= \sum_{i=1}^{n} (y_i - x_i\hat{\beta})^2.$$

- The R-square (or R^2 in general) is defined as:

$$R^2 := 1 - \frac{RSS}{Var(y_i)}.$$

- R-square measures the explanatory power of the model $x\hat{\beta}$ on the variation of y. That is to say, R^2 measures how good the fit is.

Key Statistics

- Assuming that $y = x\beta_0 + \epsilon$ is the true model. $x\hat{\beta} - x\beta_0$ is the approximation bias. $y - x\beta_0$ is the stochastic error.
- The mean-squared error (MSE):

$$MSE := \mathbb{E}[(y - x\beta_0)^2] + \mathbb{E}[(x\beta_0 - x\hat{\beta})^2].$$

- MSE accounts for both approximation bias and stochastic error even if the model is mis-specified.
- MSE is a popular measure that evaluates performance of regression like models.

In Sample and Out-of-Sample

- Typically we divide dataset into a training sample and a testing sample.
- In the training sample, we train a model by minimizing the Loss function, e.g., using linear regression. The MSE

  $$MSE_{in} := \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\hat{\beta})^2.$$

  Such metric is called in-sample MSE.
- In the testing sample $\widetilde{x}_1, \widetilde{y}_1, ..., \widetilde{x}_m, \widetilde{y}_m$, we can compute the out-of-sample MSE as:

  $$MSE_{out} := \frac{1}{n}\sum_{i=1}^{n}(\widetilde{y}_i - \widetilde{x}_i\hat{\beta})^2,$$

  by plugging in the model $\hat{\beta}$ from the training dataset.

The Effect of Model Size K

- The larger the model size K is, e.g., number of polynomials, the smaller the in-sample MSE is.
- This could lead to over fitting as K increases.
- The larger the model size K is, would the out-of-sample MSE increase or decrease?

Variance-Bias Trade off

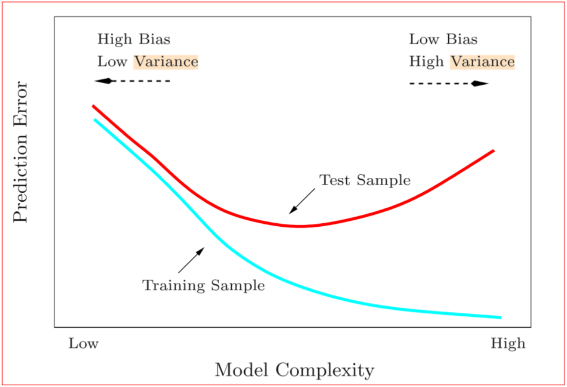


Figure 2: Bias-Variance Tradeoff

Which is your favorite Model?

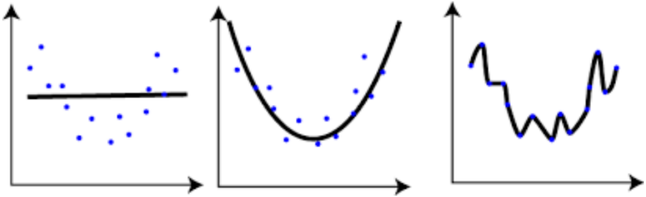
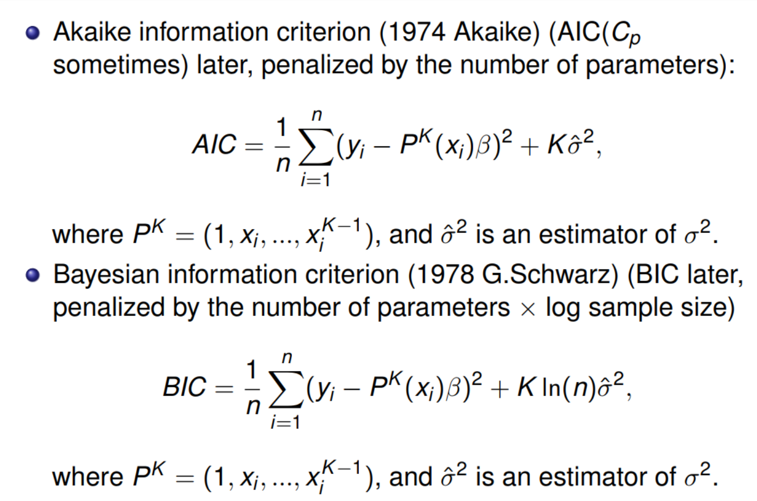


Figure 3: Model Selection

Basic Principles

- Complex model will over fit.
- Models that are too simple do not work well due to high approximation bias.
- There is a trade-off between performance and complexity.
- How do you choose in practice?

Information Criteria



- Akaike information criterion (1974 Akaike) (AIC(C_p sometimes) later, penalized by the number of parameters):

$$AIC = \frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + K\hat{\sigma}^2,$$

  where $P^K = (1, x_i, ..., x_i^{K-1})$, and $\hat{\sigma}^2$ is an estimator of σ^2.

- Bayesian information criterion (1978 G.Schwarz) (BIC later, penalized by the number of parameters × log sample size)

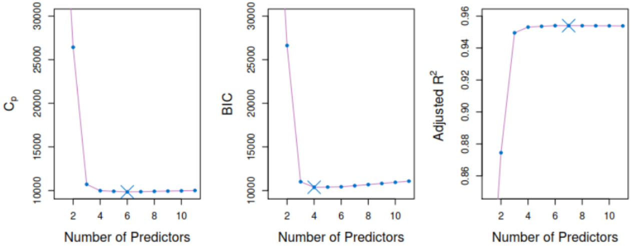
$$BIC = \frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + K\ln(n)\hat{\sigma}^2,$$

  where $P^K = (1, x_i, ..., x_i^{K-1})$, and $\hat{\sigma}^2$ is an estimator of σ^2.

What's the idea?

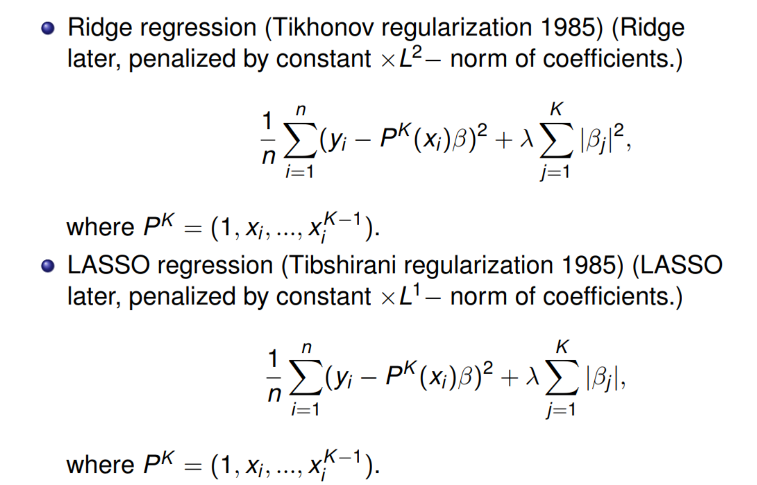
- Add a new auxiliary component on the loss function.
- This component penalizes the complexity measure K.
- The AIC/BIC criteria balances between loss function (In sample MSE) and the complexity (which affect performance in out-of-sample).

AIC/BIC



C_p
10000
15000
20000
25000
30000
2
4
6
8
10
Number of Predictors
BIC
10000
15000
20000
25000
30000
2
4
6
8
10
Number of Predictors
Adjusted R^2
0.86
0.88
0.90
0.92
0.94
0.96
2
4
6
8
10
Number of Predictors

Modern ML Approach



- Ridge regression (Tikhonov regularization 1985) (Ridge later, penalized by constant $\times L^2-$ norm of coefficients.)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + \lambda\sum_{j=1}^{K}|\beta_j|^2,$$

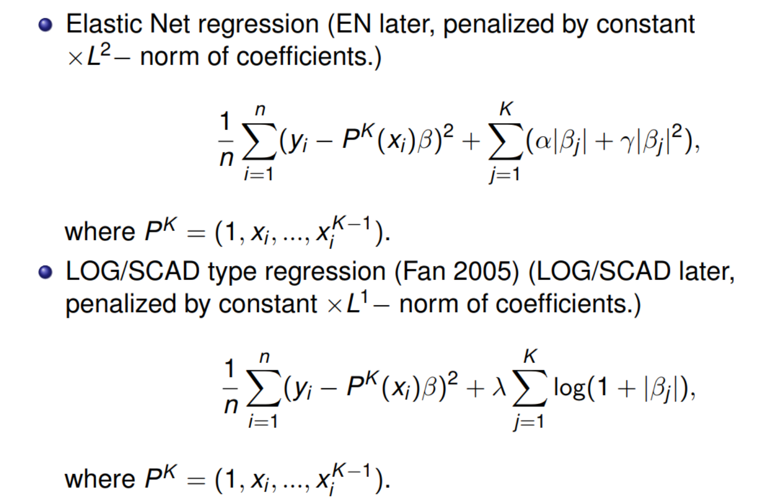
where $P^K = (1, x_i, ..., x_i^{K-1})$.

- LASSO regression (Tibshirani regularization 1985) (LASSO later, penalized by constant $\times L^1-$ norm of coefficients.)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + \lambda\sum_{j=1}^{K}|\beta_j|,$$

where $P^K = (1, x_i, ..., x_i^{K-1})$.

Modern ML Approach



- Elastic Net regression (EN later, penalized by constant $\times L^2-$ norm of coefficients.)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + \sum_{j=1}^{K}(\alpha|\beta_j| + \gamma|\beta_j|^2),$$

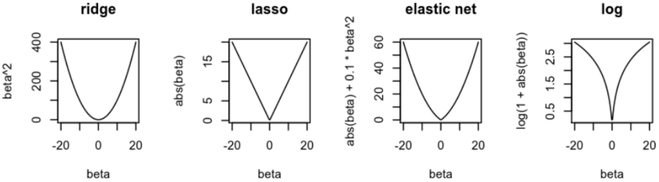
  where $P^K = (1, x_i, ..., x_i^{K-1})$.

- LOG/SCAD type regression (Fan 2005) (LOG/SCAD later, penalized by constant $\times L^1-$ norm of coefficients.)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - P^K(x_i)\beta)^2 + \lambda\sum_{j=1}^{K}\log(1 + |\beta_j|),$$

  where $P^K = (1, x_i, ..., x_i^{K-1})$.

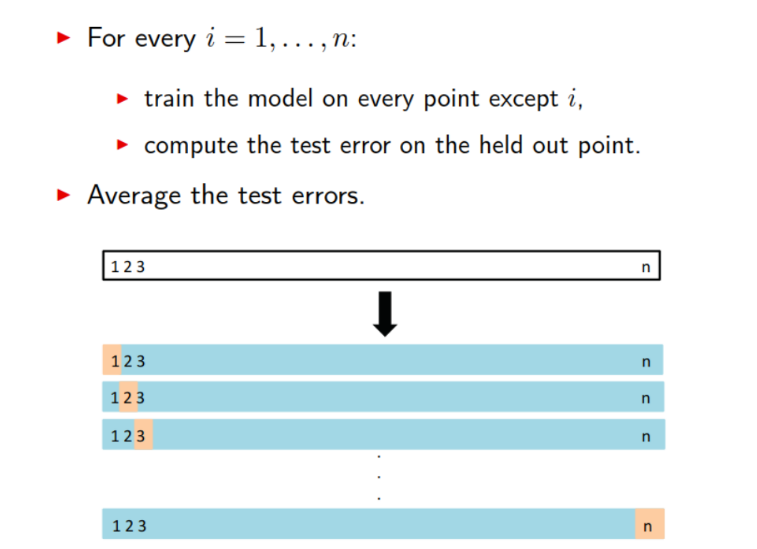
A Comparison Between ML Approaches



Why ML Needs to Adjustment beyond Regression?

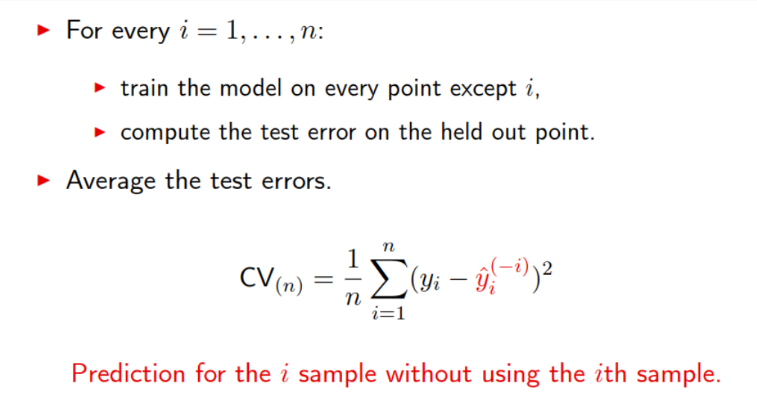
- ML is a process that determines models being used as well as the parameter in the models.
- ML needs to choose which model to use (here in the regression is the model size K).
- The adjustment of the regression chooses which model to use - Model selection.

Another Model Selection Approach: Cross-Validation



- For every $i = 1, \ldots, n$:
    - train the model on every point except i,
    - compute the test error on the held out point.
- Average the test errors.

Another Model Selection Approach: Cross-Validation

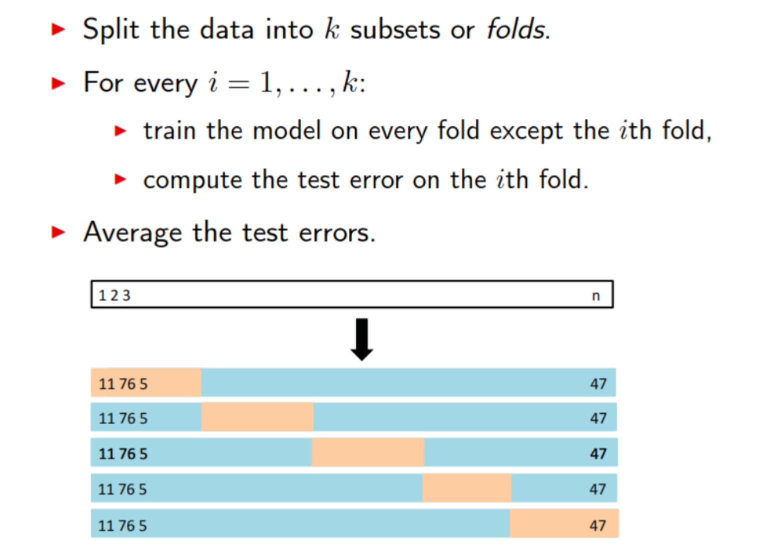


- For every $i = 1, \dots, n$:
  - train the model on every point except i,
  - compute the test error on the held out point.
- Average the test errors.

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

Prediction for the i sample without using the ith sample.

General K-Fold Cross Validation



- Split the data into k subsets or *folds*.
- For every $i = 1, \ldots, k$:
  - train the model on every fold except the ith fold,
  - compute the test error on the ith fold.
- Average the test errors.

Effect of CV

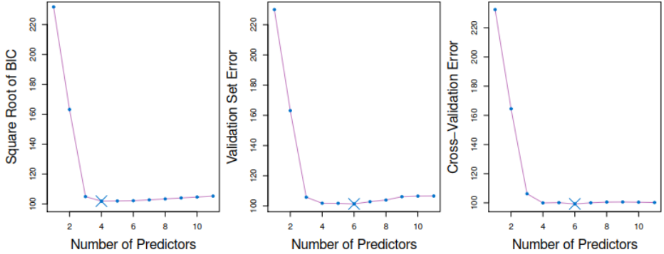


Figure 4: Effect of CV