SDSC 5001: Statistical Machine Learning I

Topic 3. Overview of Statistical Machine Learning

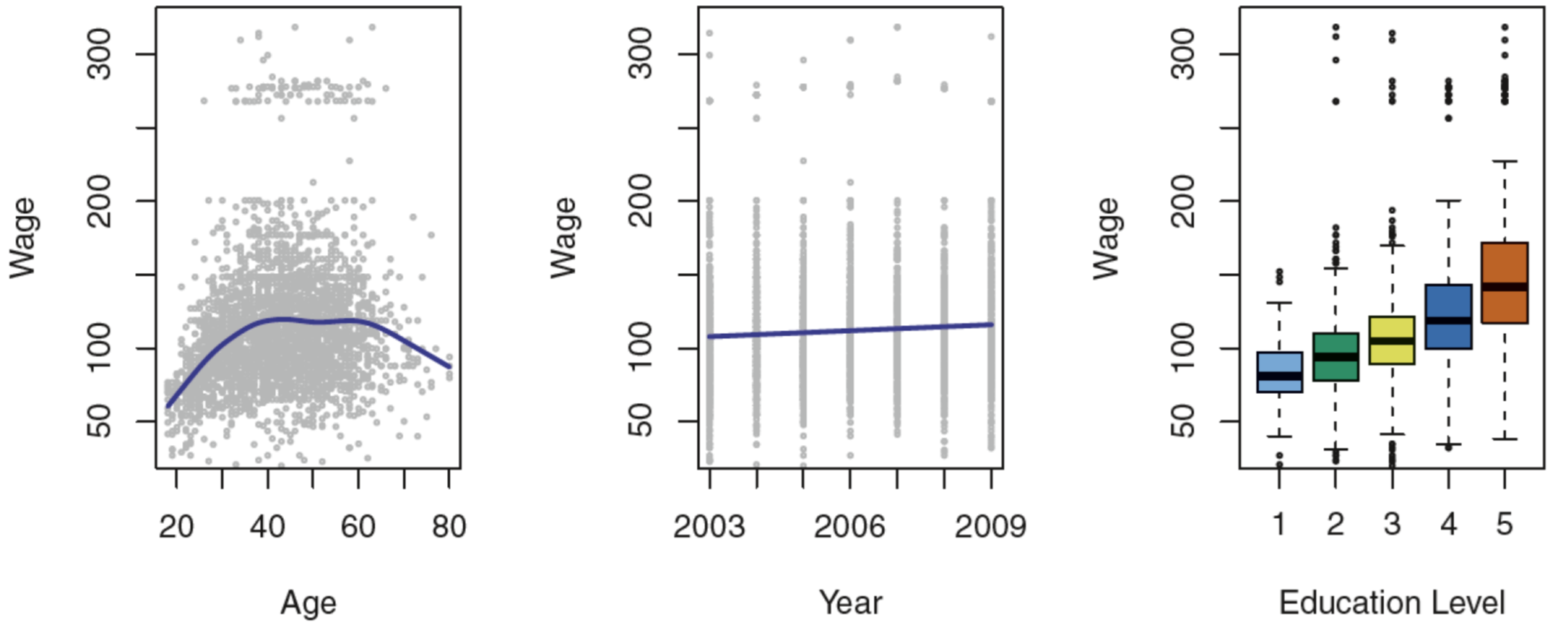
Statistics and Machine Learning

- Comparison of terminologies

Statistics	**Machine learning**
Classification/regression	Supervised learning
Clustering	Unsupervised learning
Class’n/reg’n with missing responses	Semisupervised learning
(Nonlinear) dimension reduction	Manifold learning
Covariates/responses	Features/outcomes
Sample/population	Training set/testing set
Statistical model	Learner
Misclassification/prediction error	Generalization error
Multiclass logistic function	Softmax function
Truncated linear	ReLU (rectified linear unit)
⋮	⋮

Example: Wage

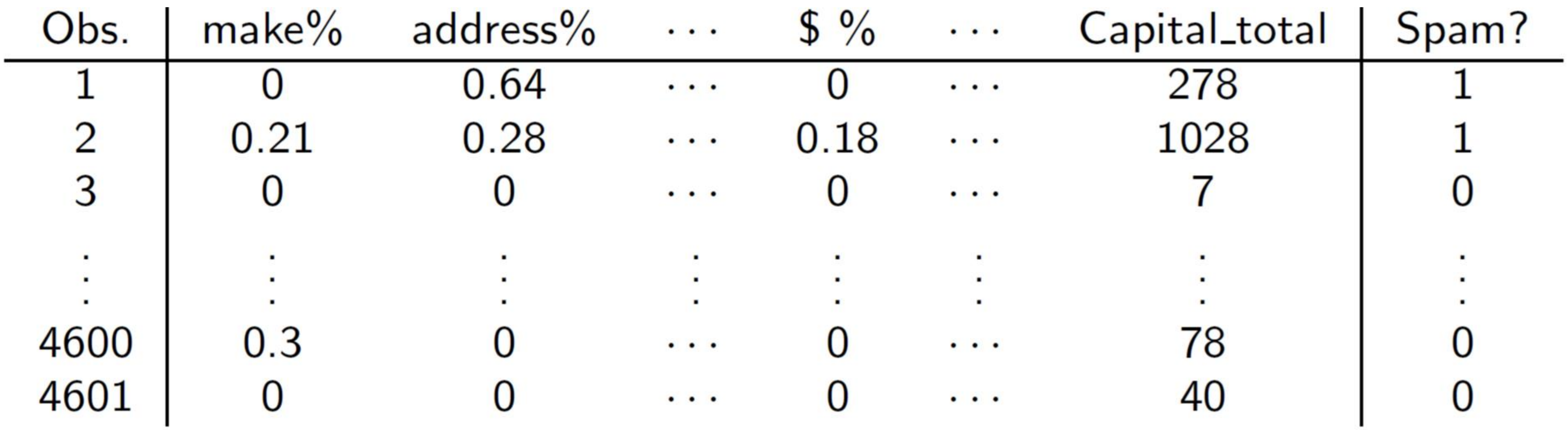
- Task: to understand the association between an employee’s wage and a number of factors



Note: the dataset was collected based on a group of males from the Atlantic region of the US.

Example: Spam Detection

- Task: To construct an email filter that can automatically detect spam emails



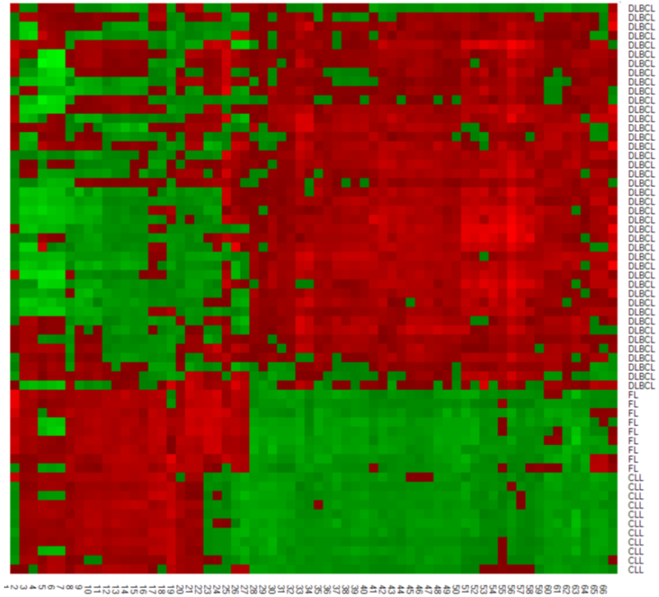
Obs.	make%	address%	···	$ %	···	Capital_total	Spam?
1	0	0.64	···	0	···	278	1
2	0.21	0.28	···	0.18	···	1028	1
3	0	0	···	0	···	7	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4600	0.3	0	···	0	···	78	0
4601	0	0	···	0	···	40	0

- The dataset has 4601 emails from 2 classes and frequencies of 57 terms.

- A trivial classification function is $I(\text{Capital\_total} > 100)$, where $I(\cdot)$ is an indicator function.

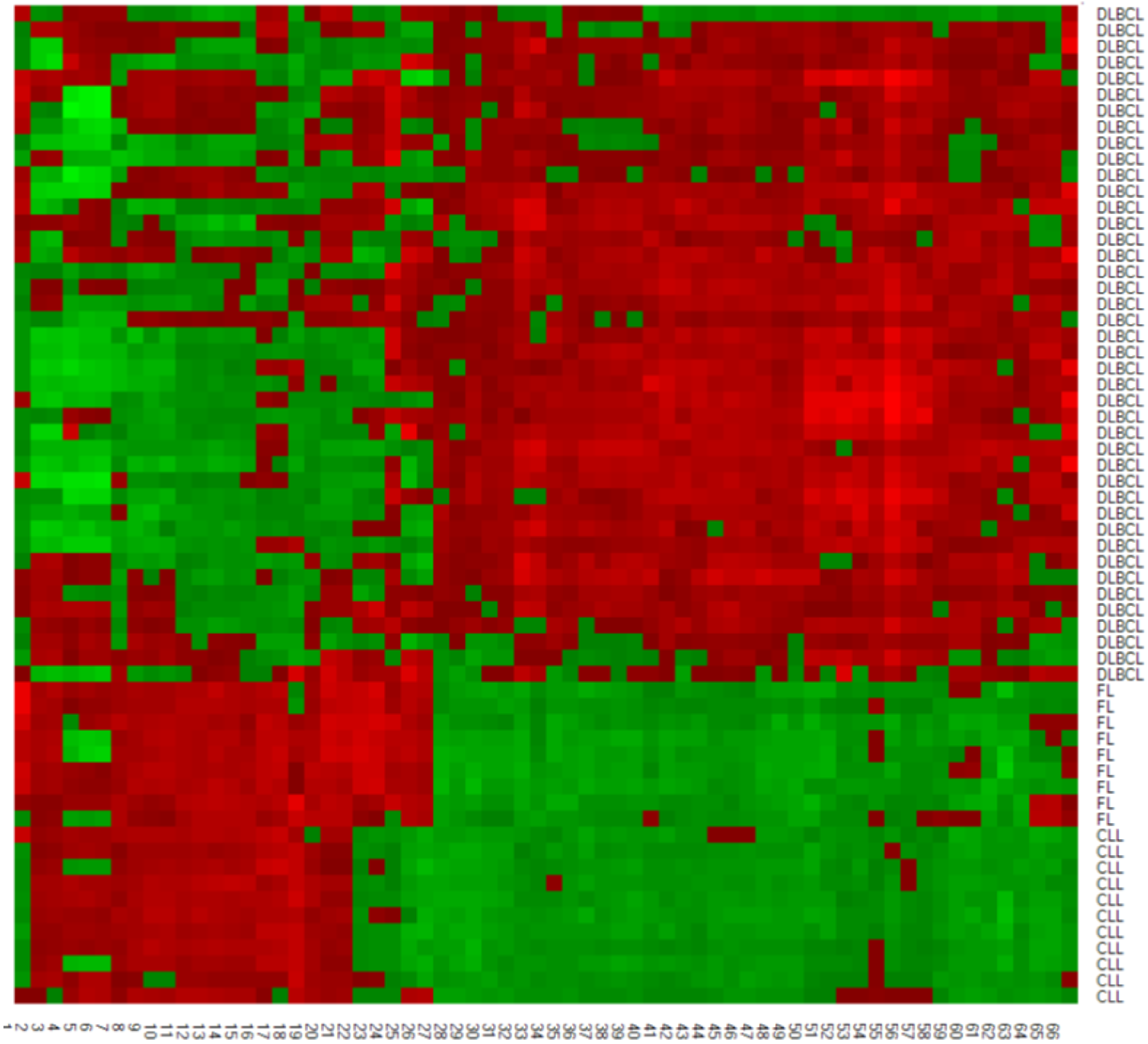
Example: Gene Microarray

- Task: to construct a classifier that can automatically diagnose cancer based on patients' genotype.



- The dataset consists of 4026 gene expression profiles of 62 patients from 3 types of adult lymphoid malignancies.
- Displayed are 66 "carefully" selected genes

Example: Gene Microarray



Notations

- A training sample $(x_i, y_i)_{i=1}^{n}$ is available, where
  - x_i: inputs, feature vectors, predictors, independent variables

    $x_i \in R^p$; qualitative features are coded using, for example, dummy variables
  - y_i: output, response, dependent variable

    y_i is a scalar, and it can be a real vector in some cases.
  - In general, we assume the data are generated from

$$Y = f(X) + \epsilon$$

    - f is some unknown function representing the systematic information that X provides about Y.
    - ϵ is a random error with mean 0 and independent of X.

Estimation

- Data analysis is all about modeling and estimation of f based on the training sample.

- Parametric models
  - Linear/polynomial regression model
  - Generalized linear regression model
  - Fisher's discriminant analysis
  - Logistic regression
  - Deep learning

- Nonparametric models
  - Local smoothing
  - Smoothing splines
  - Classification and regression trees; random forest; boosting
  - Support vector machines

Prediction and Inference

- **Prediction:** based on an estimate of f, denoted as $\hat{f}$, we can predict the response for any new X as

$$\hat{Y} = \hat{f}(X)$$

  - The exact form of $\hat{f}$ is not a concern.
  - The prediction error consists of a **reducible error** and an **irreducible error**:

$$E\left(\hat{Y} - Y\right)^2 = E\left(\hat{f}(X) - f(X) - \epsilon\right)^2 = E\left(\hat{f}(X) - f(X)\right)^2 + \mathrm{var}(\epsilon)$$

  - The reducible error can be potentially improved by using appropriate learning technique to estimate f.
  - The irreducible error $\mathrm{var}(\varepsilon)$ is due to the fact that ε cannot be predicted using X.

Prediction and Inference

- **Inference:** to understand the way how Y is affected by X, or how Y changes as a function of X.
  - Which predictors are associated with Y?
  - What is the relationship between Y and each predictor?
  - What is the change of Y if we intervene certain predictor?

- **Balance between prediction and inference**
  - Some simple models, say linear models, allow for clear interpretable inference, but may not yield accurate predictions.
  - Highly nonlinear models can potentially provide accurate prediction, for which inference is yet more challenging.

Classification

- Classification is slightly different from regression, where

$$P(Y=k|X)=f_k(X); k=1,\dots,K$$

- After $\hat{f}_k$ is obtained, we set the classification decision function

$$\hat{\phi}(X)=\underset{k}{\operatorname{argmax}}\,\hat{f}_k(X)$$

- The prediction (misclassification) error is

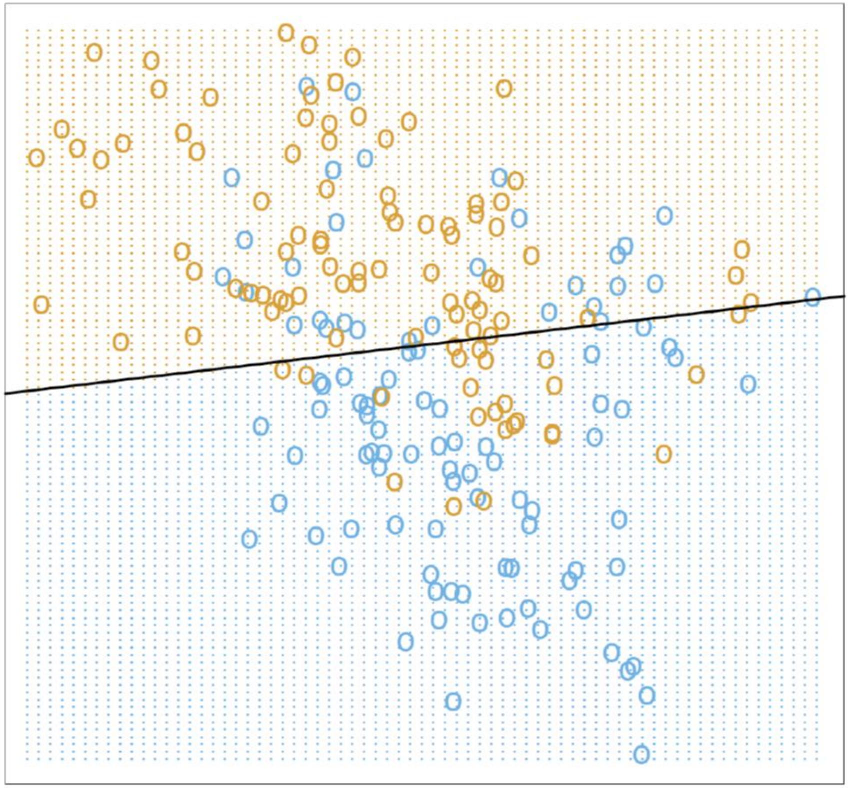
$$P\big(Y\neq\hat{\phi}(X)\big)=E\left(I\big(Y\neq\hat{\phi}(X)\big)\right)$$

Classification

$$P\left(Y \neq \hat{\phi}(X)\right) = E\left(I\left(Y \neq \hat{\phi}(X)\right)\right)$$

A Toy Example

- 200 points simulated from an unknown distribution; 100 in each of two classes {BLUE, ORANGE}. Can we build a rule to predict the color of future points?



Model 1: Linear Regression

- Code $Y = 1$ if ORANGE, else $Y = 0$.
- Model Y as a linear function of X,

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j = \mathbf{X}\beta$$

- Obtain β by least squares estimation (LSE),

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

  - Simple algebra implies that $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\, \mathbf{X}^T\mathbf{y}$.
  - The classification decision function is $\hat{\phi}(X) = I\left(X^T\hat{\beta} > 0.5\right)$.

Model 2: Nearest Neighbors

- A natural way to predict a future point based on its neighbors,

$$\hat{y}(X) = \frac{1}{k}\sum_{i=1}^{n} y_i I(x_i \in N_k(X))$$

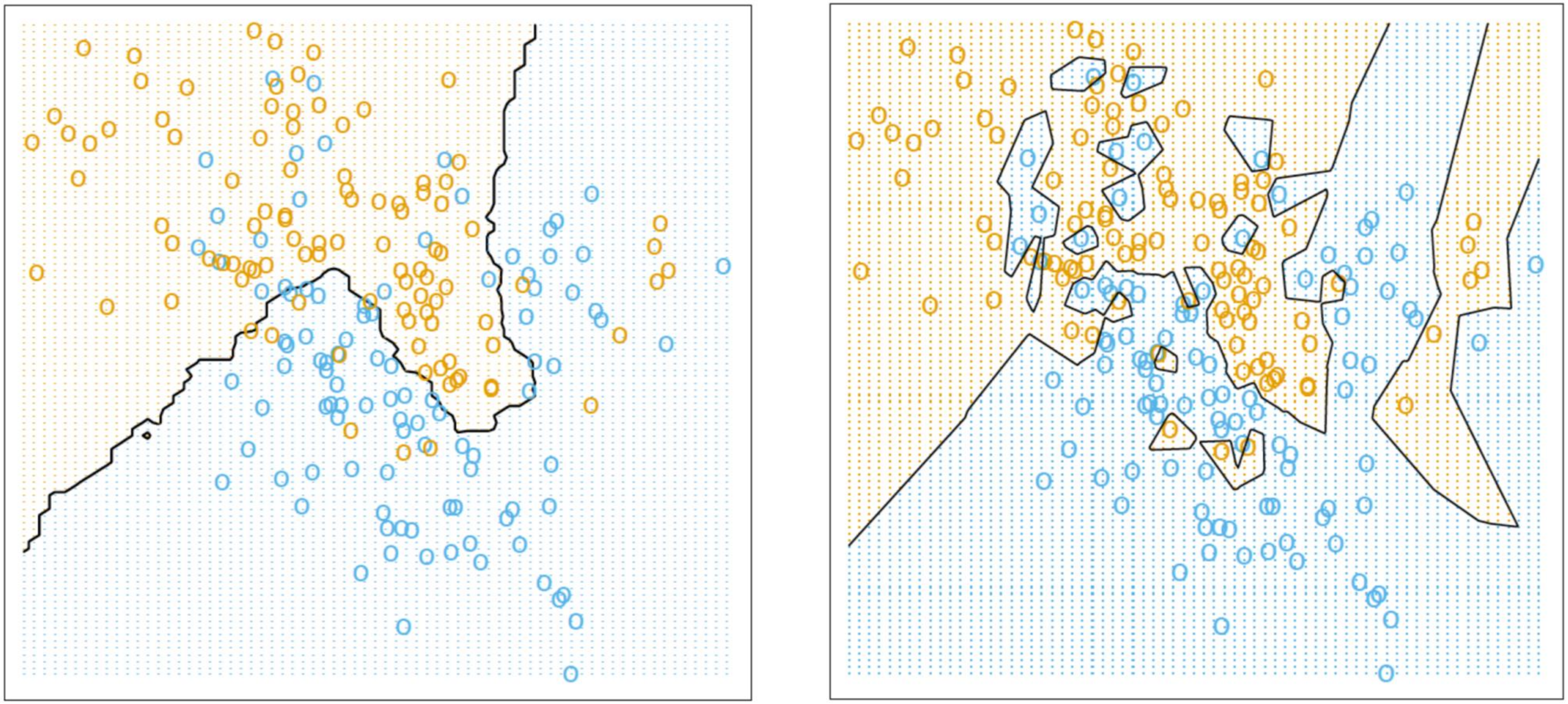
where $N_k(X)$ is a neighborhood of X that contains exactly k neighbors (k-nearest neighbors).

- If one class has a clear dominance in $N_k(X)$, then it is likely that X itself would belong to that class too.

- Thus the classification decision function is a majority voting among the members of $N_k(X)$,

$$\hat{\phi}(X) = I(\hat{y}(X) > 0.5)$$

Nearest Neighbors



The Figures

- Left panel shows the result of 15-NN classifier. A few training data are misclassified, and the decision boundary adapts to the local density of the classes.

- Right panel shows the result of 1-NN classifier. None of the training data is misclassified.

- Linear regression uses 3 parameters to describe its model. How many does k-NN classifier use?

- In fact, k-NN classifier uses n/k effective number of parameters (need advanced techniques)

Model Assessment for Regression

- Assume $Y \in R$, $X \in R^p$. The accuracy of f can be measured by its mean square error (MSE)

$$MSE(f) = E(Y - f(X))^2$$

where E is taken with respect to both X and Y.

- The minimizer of $MSE(f)$ is $f^*(X) = E(Y|X)$.

- As the distribution of (X, Y) is unknown, $MSE(f)$ needs to be approximated based on the sample data.

Training Error

- Given a training sample $(x_i, y_i)_{i=1}^{n}$ and the estimated function $\hat{f}$, its training error is given by

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

- But we do not really care how well the model works on the training data.
- Training error tends to under-estimate $MSE(f)$.
- It is possible to build a complicated model with small training error, or even zero training error.

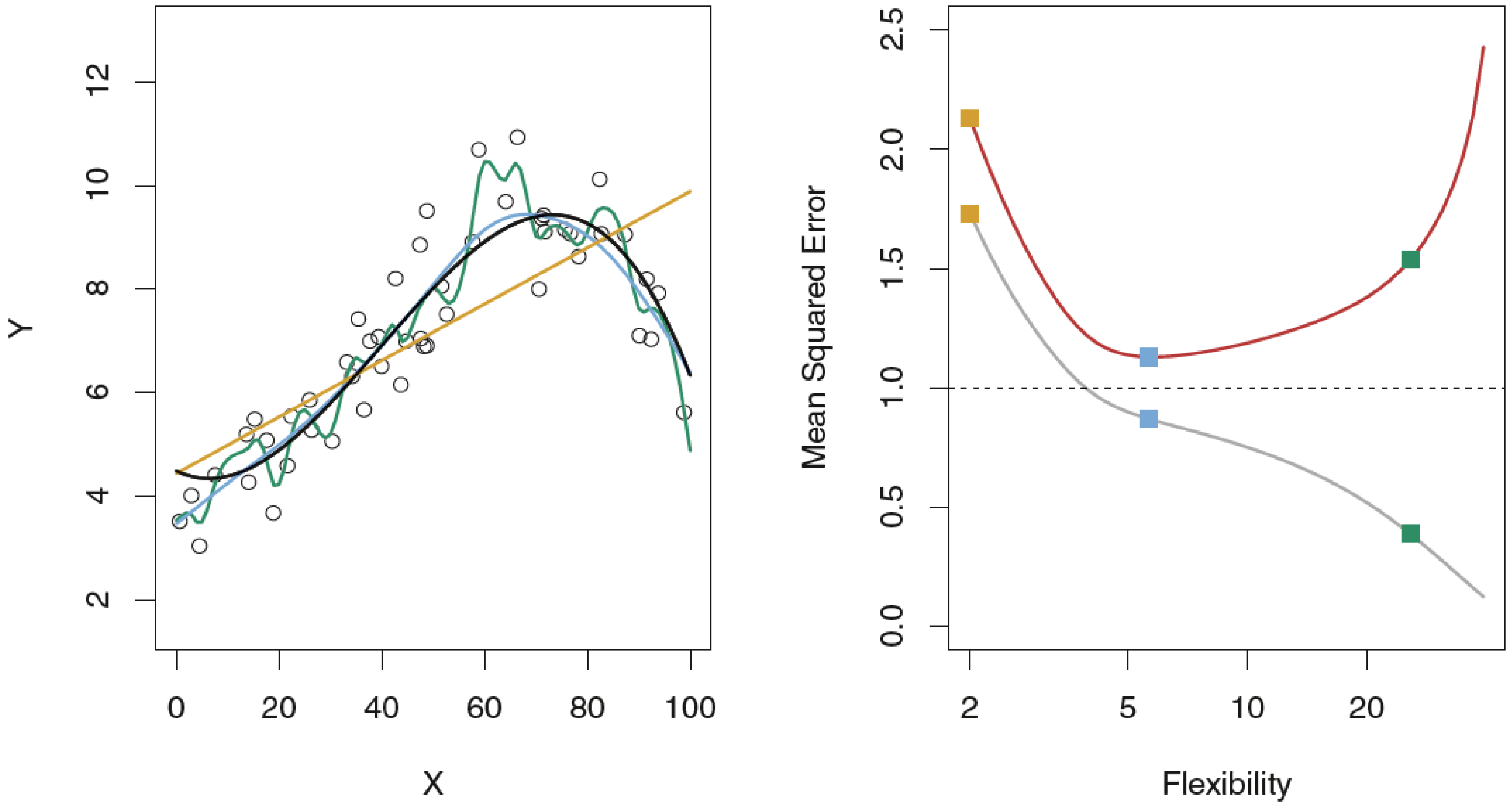
Test Error

- A better estimate of $MSE(f)$ should be obtained based on a new test sample that are independent of the training sample.
- Assume we had a test sample $(x_{0i}, y_{0i})_{i=1}^{m}$, then the test error of $\hat{f}$ is given by

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_{0i} - \hat{f}(x_{0i})\right)^2$$

- In general, test error is closer to $MSE(f)$ than training error.
- The test sample are not seen during training, and mimic the future observations that we attempt to predict.

Training/Test Errors of Smoothing Spline



Bias-Variance Decomposition

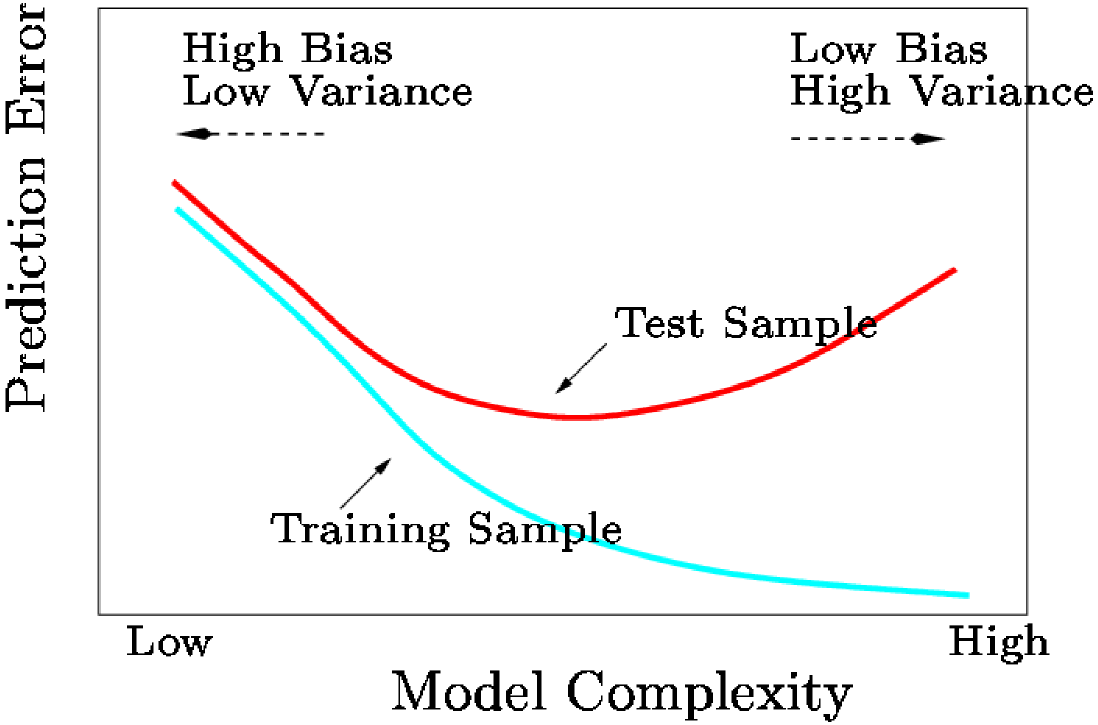
- The U-shape of the test error turns out to be the result of two competing quantities

$$E\left(Y-\hat{f}(X)\right)^2 = E\left(\hat{f}(X)-f(X)\right)^2 + \mathrm{var}(\varepsilon)$$
$$= \left(Bias\left(\hat{f}(X)\right)\right)^2 + \mathrm{var}\left(\hat{f}(X)\right) + \mathrm{var}(\varepsilon)$$

- $Bias\left(\hat{f}(X)\right) = E\left(\hat{f}(X)\right) - f(X)$ refers to the error that is introduced by approximating f.
- $\mathrm{var}\left(\hat{f}(X)\right) = E\left(\hat{f}(X) - E\left(\hat{f}(X)\right)\right)^2$ refers to how much $\hat{f}$ would change if we estimated it using a different training set.
- In general, as we use more complicated methods, the variance will increase and the bias will decrease.

A Fundamental Picture

- In general, training errors always decline as flexibility (complexity) increases.
- However, test error shows a **U-shape**:
  - decline at first (as reduction in bias dominates)
  - then start to increase again (as increase in variance dominates).



Model Assessment for Classification

- Assume $Y \in \{1, \dots, K\}$ and $X \in R^p$. The accuracy of f can be measured by its misclassification error,

$$MCE(f) = E\big(I(Y \neq f(X))\big)$$

where E is taken with respect to both X and Y.

- The minimizer of $MCE(f)$ must satisfy

$$f^*(X) = \arg\max_k P(Y = k|X)$$

which is also known as the Bayes rule.

Training and Test Errors

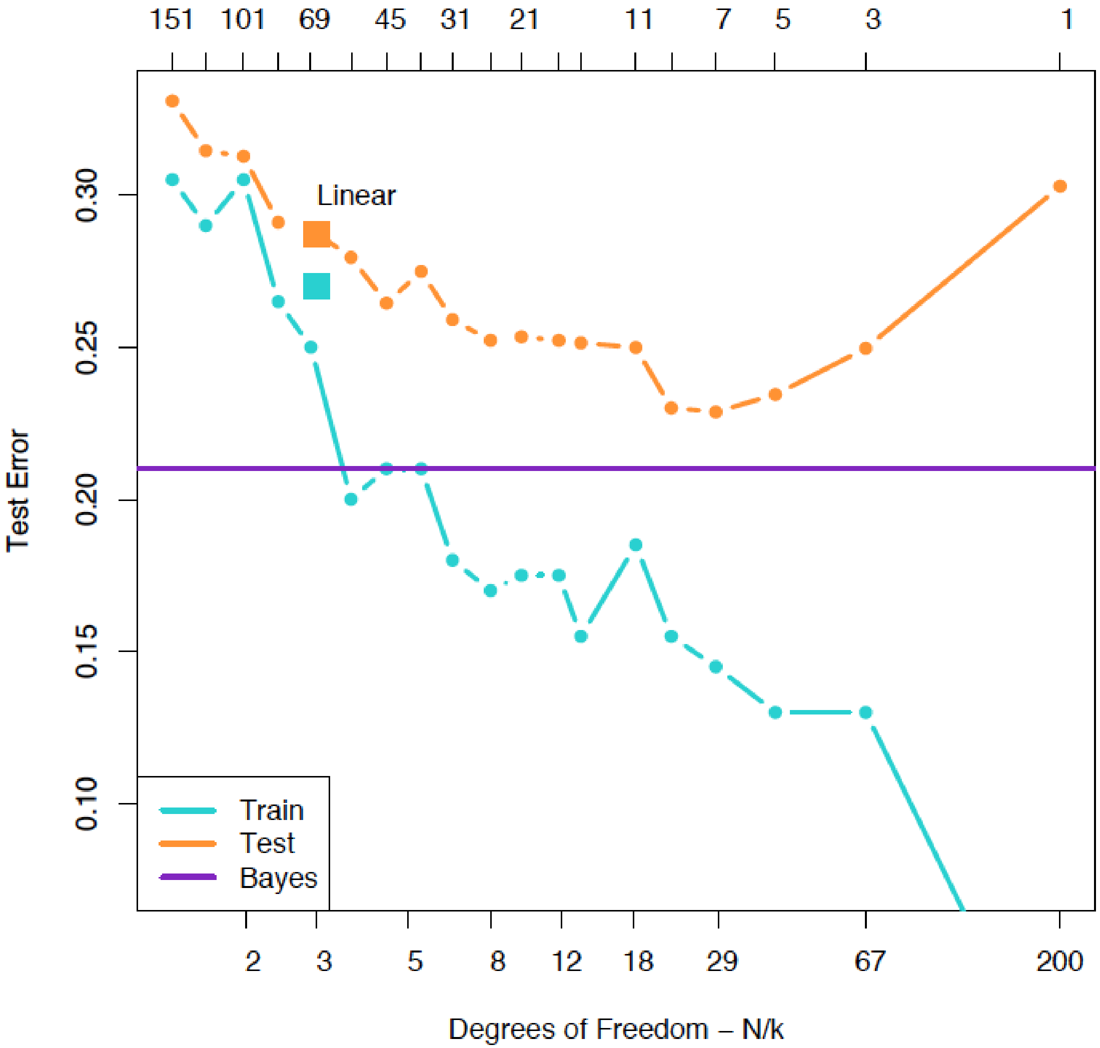
- Given a training sample $(x_i, y_i)_{i=1}^{n}$ and the estimated function $\hat{f}$, its training error is given by

$$\frac{1}{n}\sum_{i=1}^{n} I\big(y_i \neq \hat{f}(x_i)\big)$$

- If we had a test sample $(x_{0i}, y_{0i})_{i=1}^{m}$, then the test error of $\hat{f}$ is given by

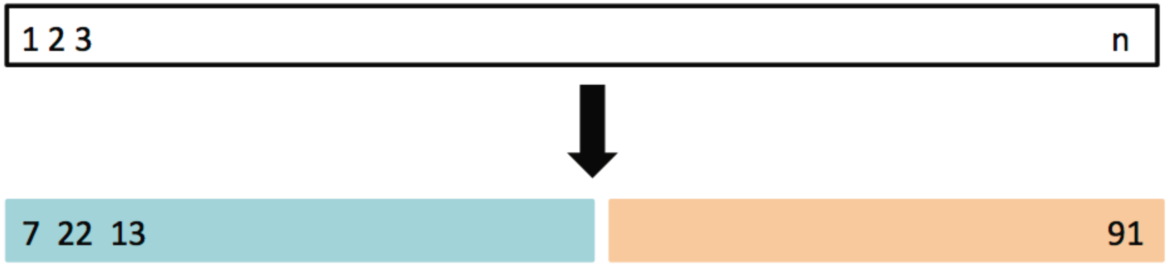
$$\frac{1}{m}\sum_{i=1}^{m} I\big(y_{0i} \neq \hat{f}(x_{0i})\big)$$

Training/Test Errors of KNN



Test Error in Practice

- In practice, a large designated test set is often unavailable.
- Some adjustments can be made to the training error in order to estimate the test error: AIC, BIC, Covariance penalty.
- If we have a large training set, we can estimate the test error by randomly splitting the data into training and validation parts.



Use the training part to build model, and then assess the model by applying it to the validation part.

Validation Set Approach

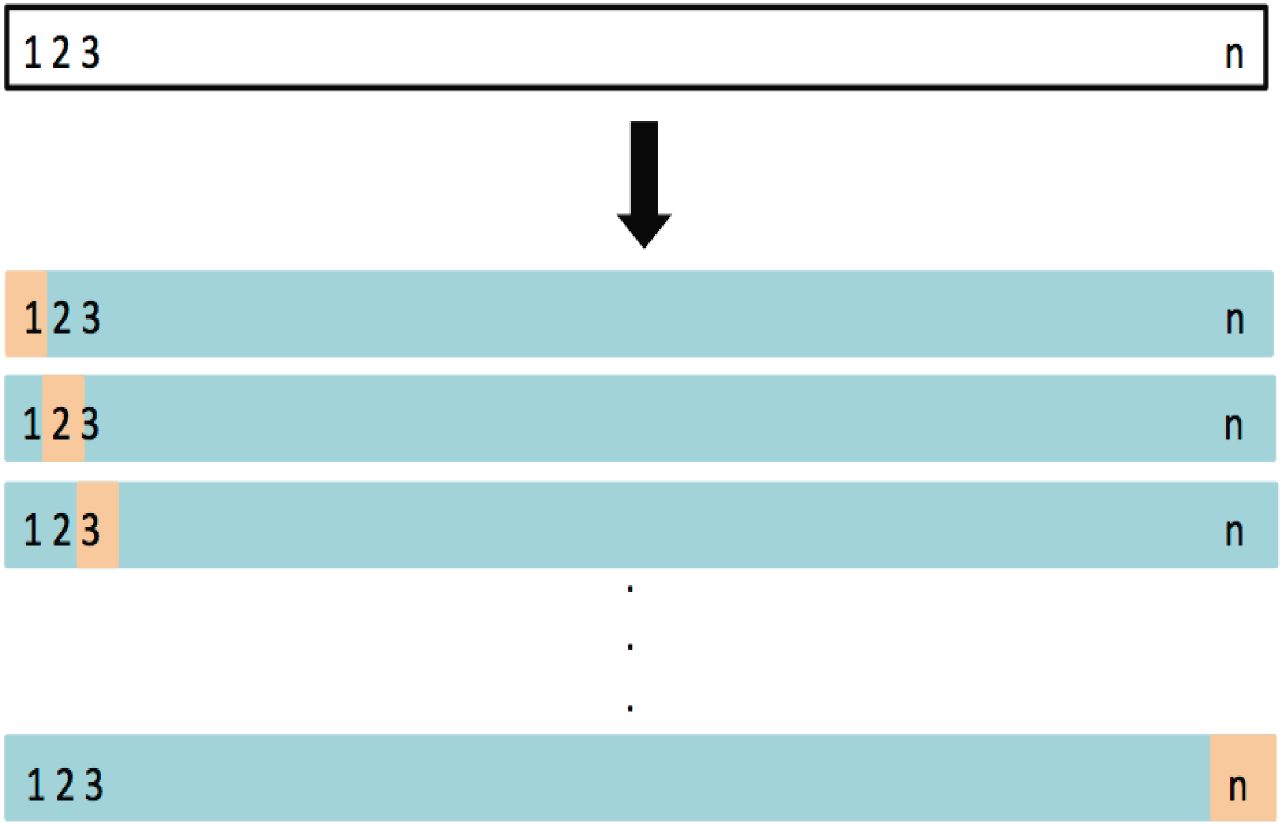
- Advantages
  - Simple idea
  - Easy to implement

- Disadvantages
  - The validation MSE can be highly variable.
  - Only a subset of observations are used to fit the model, while statistical methods tend to perform worse when trained on fewer observations.

Leave-One-Out Cross Validation (LOOCV)

- This method is similar to the validation set approach, with its disadvantages addressed.
- Split the dataset of size n into
  - Training set with size $n-1$
  - Validation set with size 1
- Fit the model using the training set.
- Validate model using the validation set, and compute the corresponding MSE.
- Repeat this process n times.
- The MSE for the model is computed as the averaged MSE’s.

LOOCV



LOOCV vs. Validation Set Approach

- LOOCV has less bias as the training data contains $n-1$ observations, i.e., almost the whole data set.

- LOOCV produces a less variable MSE.

- LOOCV is computationally intensive (disadvantage).

K-Fold CV

- Divide the training sample into $A_1, \ldots, A_K$ ($K = 2, 5, 10$ or n)

- For each k, fit the model to $\{A_1, \ldots, A_{k-1}, A_{k+1}, \ldots, A_K\}$, giving the estimated model $\hat{f}^{-k}(x)$.

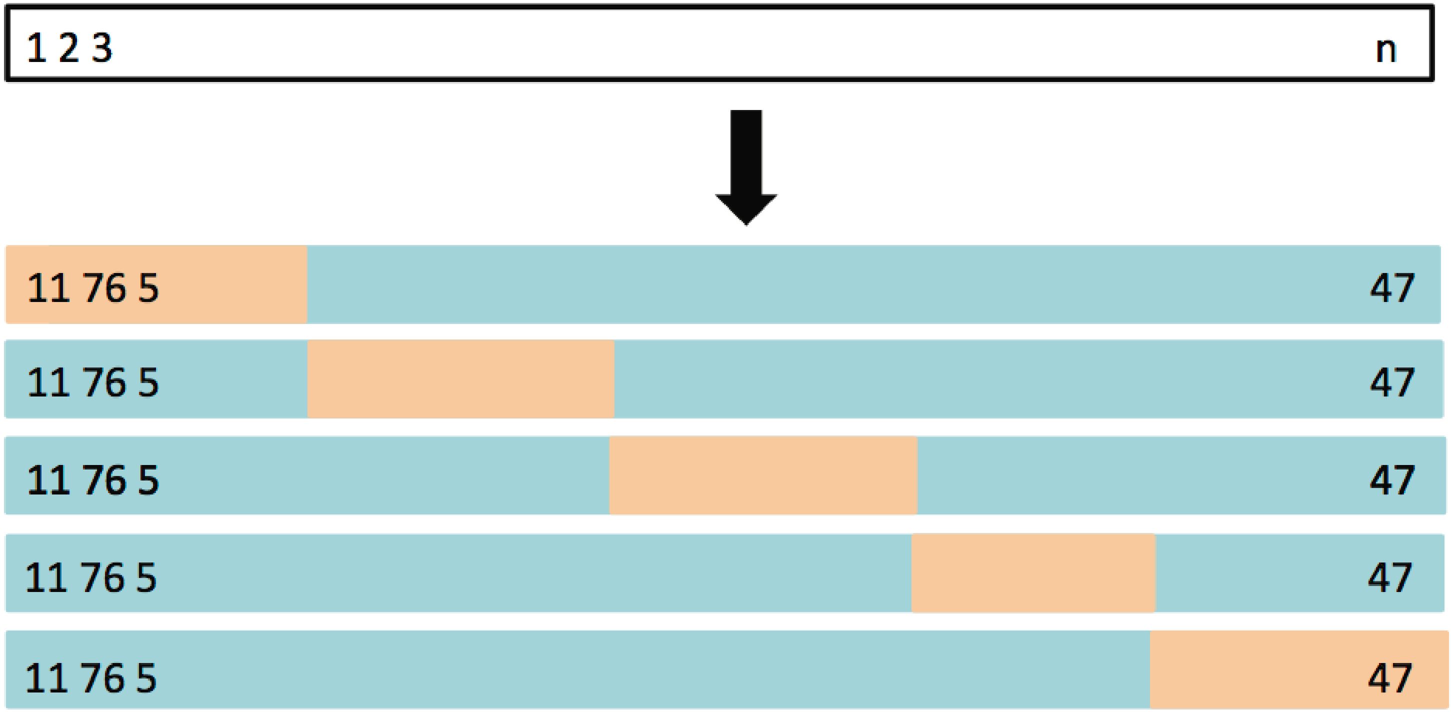
- Compute its prediction error on A_k,

$$E_k(\hat{f}) = \sum_{i \in A_k} L\left(y_i, \hat{f}^{-k}(x_i)\right)^2$$

- Aggregating all $E_k(\hat{f})$'s gives the CV error

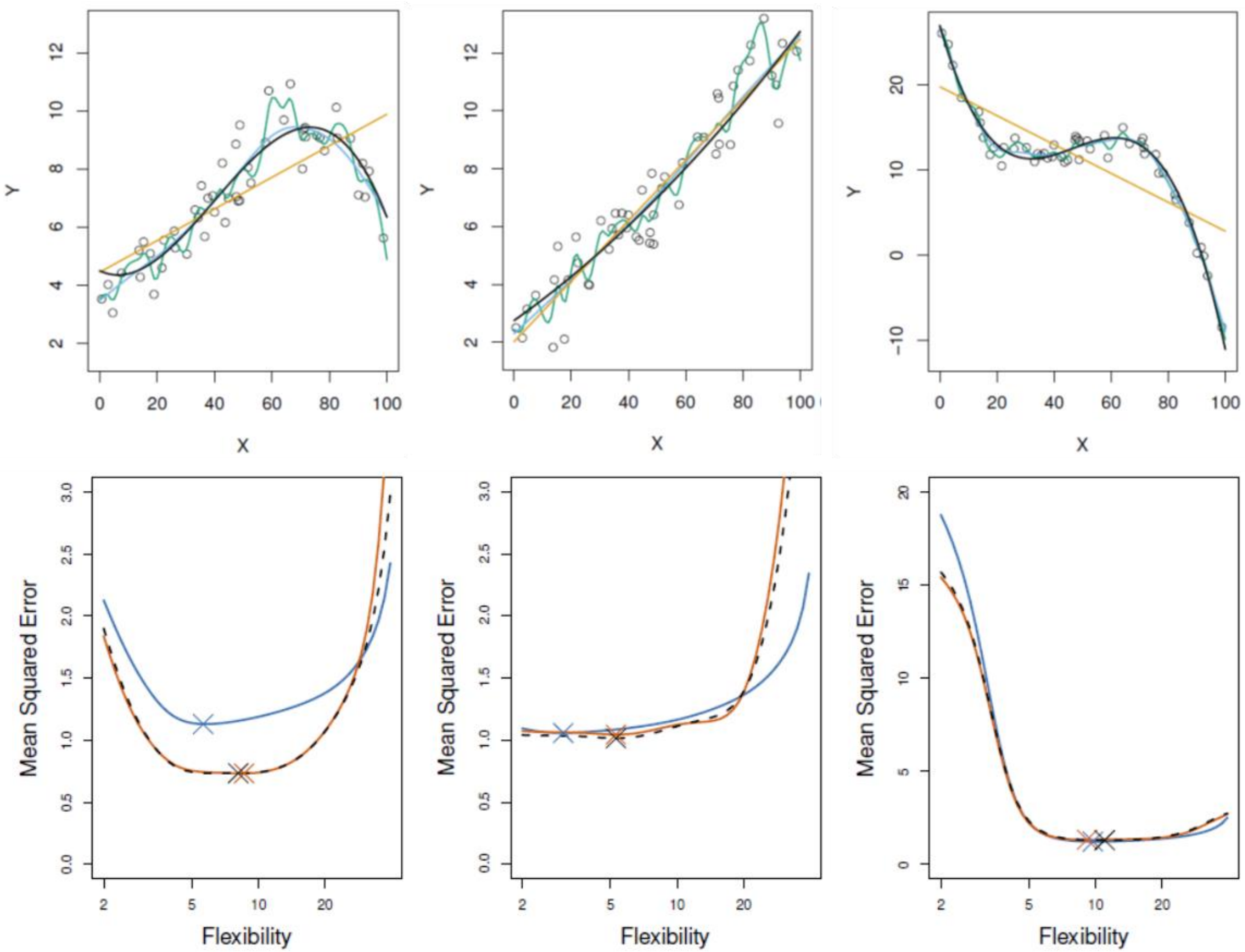
$$CV(\hat{f}) = \frac{1}{n} \sum_{k=1}^{K} E_k(\hat{f})$$

Cross Validation



Three Simulated Examples

- Blue: test error; Black: LOOCV; Orange: 10-fold CV



Comments

- Which is better, LOOCV or K-fold CV?
  - LOOCV is less biasd than K-fold CV (when $K < n$).
  - But LOOCV has higher variance than K-fold CV (when $K < n$)
  - Thus there is a trade-off between what to use.

- Conclusion
  - We tend to use K-fold CV with $K = 5$ or $K = 10$.
  - It has been empirically shown that they yield reasonable estimates of test error.