Principles of Statistical Machine Learning
Basics of k Nearest Neighbors Regression Learning

Ernest Fokoué
福尔特　教授

School of Mathematical Sciences
Rochester Institute of Technology
Rochester, New York, USA

Principles of Statistical Machine Learning
STAT 747-Autumn Semester 2017

Regression Analysis Dataset

rating	complaints	privileges	learning	raises	critical	advance
43	51	30	39	61	92	45
63	64	51	54	63	73	47
71	70	68	69	76	86	48
61	63	45	47	54	84	35
81	78	56	66	71	83	47
43	55	49	44	54	49	34
58	67	42	56	66	68	35
71	75	50	55	70	66	41
72	82	72	67	71	83	31
67	61	45	47	62	80	41
64	53	53	58	58	67	34
67	60	47	39	59	74	41
69	62	57	42	55	63	25

Build a learning machine that has a low prediction error on *rating*.

```
head(attitude)
```

Regression Analysis Dataset

lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa
-0.58	2.77	50	-1.39	0	-1.39	6	0	-0.43
-0.99	3.32	58	-1.39	0	-1.39	6	0	-0.16
-0.51	2.69	74	-1.39	0	-1.39	7	20	-0.16
-1.20	3.28	58	-1.39	0	-1.39	6	0	-0.16
0.75	3.43	62	-1.39	0	-1.39	6	0	0.37
-1.05	3.23	50	-1.39	0	-1.39	6	0	0.77
0.74	3.47	64	0.62	0	-1.39	6	0	0.77
0.69	3.54	58	1.54	0	-1.39	6	0	0.85
-0.78	3.54	47	-1.39	0	-1.39	6	0	1.05
0.22	3.24	63	-1.39	0	-1.39	6	0	1.05
0.25	3.60	65	-1.39	0	-1.39	6	0	1.27
-1.35	3.60	63	1.27	0	-1.39	6	0	1.27

Build a learning machine that has a low prediction error on lpsa.

```
library(ElemStatLearn); data(prostate)
```

Motivating Example Regression Analysis

Consider the univariate function $f \in \mathcal{C}([0, 2\pi])$ *given by*

$$f(\mathbf{x}) = \frac{\pi}{2}\mathbf{x} + \frac{3}{4}\pi \cos\left\{\frac{\pi}{2}(1+\mathbf{x})\right\} \tag{1}$$

Simulate of an artificial iid data set $\mathcal{D} = \{(\mathbf{x}_i, \mathrm{y}_i),\ i = 1, \cdots, n\}$, *with*

- $n = 99$ *and* $\sigma = \pi/3$
- $\mathbf{x}_i \in [0, 2\pi]$ *drawn deterministically and equally spaced*
- $Y_i = f(\mathbf{x}_i) + \varepsilon_i$
- $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

The R code is

```
f <- function(x){(pi/2)*x + (3*pi/4)*cos((pi/2)*(1+x))}
x <- seq(0, 2*pi, length=n)
y <- f(x) + rnorm(n, 0, pi/3)
```

Motivating Example Regression Analysis

Consider the univariate function $f \in \mathcal{C}([-1,+1])$ *given by*

$$f(\mathbf{x}) = -\mathrm{x} + \sqrt{2}\sin(\pi^{3/2}\mathrm{x}^2) \tag{2}$$

Simulate of an artificial iid data set $\mathcal{D} = \{(\mathbf{x}_i, \mathrm{y}_i),\ i = 1, \cdots, n\}$, *with*

- $n = 99$ *and* $\sigma = 3/10$
- $\mathbf{x}_i \in [-1,+1]$ *drawn deterministically and equally spaced*
- $Y_i = f(\mathbf{x}_i) + \varepsilon_i$
- $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

The R code is

```
f <- function(x){-x + sqrt(2)*sin(pi^(3/2)*x^2)}
x <- seq(-1, +1, length=n)
y <- f(x) + rnorm(n, 0, 3/10)
```

Risk Functionals and Cost Functions

1. *Definition of a risk functional,*

$$R(f) = \mathbb{E}[\ell(Y, f(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(\mathrm{y}, f(\mathrm{x}))p_{XY}(\mathrm{x},\mathrm{y})d\mathrm{x}d\mathrm{y}$$

   $R(f)$ *is the expected loss over all pairs of the cross space* $\mathcal{X} \times \mathcal{Y}$.

2. *Ideally, one seeks the best out of all possible functions, i.e.,*

$$f^*(X) = \underset{f}{\mathtt{arg\,min}}\, R(f) = \underset{f}{\mathtt{arg\,min}}\, \mathbb{E}[\ell(Y, f(X))]$$

   $f^*(\cdot)$ *is such that*

$$R^* = R(f^*) = \min_f R(f)$$

3. *This ideal function cannot be found in practice, because the fact that the distributions are unknown, make it impossible to form an expression for* $R(f)$.

Cost Functions and Risk Functionals

- *Theorem: Under regularity conditions,*

$$f^*(X) = \mathbb{E}[Y|X] = \underset{f}{\texttt{argmin}}\,\mathbb{E}[(Y - f(X))^2]$$

  Under the squared error loss, the optimal function f^* that yields the best prediction of Y given X is no other than the expected value of Y given X.

- *Since we know neither $p_{XY}(\mathrm{x},\mathrm{y})$ nor $p_X(\mathrm{x})$, the conditional expectation*

$$\mathbb{E}[Y|X] = \int_{\mathcal{Y}} \mathrm{y} p_{Y|X}(\mathrm{y})(d\mathrm{y}) = \int_{\mathcal{Y}} \mathrm{y} \frac{p_{XY}(\mathrm{x},\mathrm{y})}{p_X(\mathrm{x})} d\mathrm{y}$$

  cannot be directly computed.

kNearest Neighbors Regression Learning

Basic Ideas and Concepts

Intuition of k-Nearest Neighbor Regression

- *k-Nearest Neighbor Principle: A reasonable prediction of the response value for a given object is the average of the response values of its nearest neighbors*

- *k-Nearest Neighbor Steps: Given a new point for which a predicted response is needed,*
    - *Choose a distance for measuring how far a given point is from another*
    - *Set the size of the neighborhood k*
    - *Compute the distance from each existing point to the new point*
    - *Identify the response values of the k points closest/nearest to the new point*
    - *Compute the average of the response values of those k neighbors as the best estimate of the response value of the new point*

- *k-Nearest Neighbor Regression: The estimated response value of a vector $\boldsymbol{x}$ is the average of the response values in the neighborhood of $\boldsymbol{x}$.*

k-Nearest Neighbors (kNN) regression

$\mathcal{D} = \Big\{(\mathbf{x}_1, Y_1), \cdots, (\mathbf{x}_n, Y_n)\Big\}$, *with* $\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{iq})^\top \in \mathcal{X}^q$, $Y_i \in \mathbb{R}$.

1. *Choose the value of k and the distance to be used*
2. *Let $\mathbf{x}^*$ be a new point. Compute*

$$d_i^* = d(\mathbf{x}^*, \mathbf{x}_i) \qquad i = 1, \cdots, n$$

3. *Rank all the distances d_i^* in increasing order*

$$d_{(1)}^* \leq d_{(2)}^* \leq \cdots \leq d_{(k)}^* \leq d_{(k+1)}^* \leq \cdots \leq d_{(n)}^*$$

4. *Form $\mathcal{V}_k(\mathbf{x}^*)$, the k-Neighborhood of $\mathbf{x}^*$*

$$\mathcal{V}_k(\mathbf{x}^*) = \{\mathbf{x}_i : \; d(\mathbf{x}^*, \mathbf{x}_i) \leq d_{(k)}^*\}$$

5. *Compute the predicted response $\hat{Y}^*$ as*

$$\hat{Y}_{\mathtt{kNN}}^* = \hat{f}_{\mathtt{kNN}}(\mathbf{x}^*) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x}^*)} Y_i = \frac{1}{k} \sum_{i=1}^{n} Y_i I(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x}^*))$$

Some Distances Used in kNN Regression

Three of the most commonly used distances are:

- Euclidean distance: *also known as the ℓ_2 distance*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{l=1}^{q} (\mathrm{x}_{il} - \mathrm{x}_{jl})^2} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$$

- Manhattan distance (city block): *also known as ℓ_1 distance*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{l=1}^{q} |\mathrm{x}_{il} - \mathrm{x}_{jl}| = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1$$

- Maximum distance: *also known as the infinity distance*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_{l=1,\cdots,q} |\mathrm{x}_{il} - \mathrm{x}_{jl}| = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_\infty$$

Other distances: Minkowski*;* canberra*;* binary *or* Jaccard*.*

Additional Distances for kNearest Neighbors

Given two vectors $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^q$, The most common distances are

- *Minkowski distance: ℓ_p distance*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\{ \sum_{\ell=1}^{q} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|^p \right\}^{1/p}$$

- *Canberra distance:*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{\ell=1}^{q} \frac{|\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|}{|\mathrm{x}_{i\ell} + \mathrm{x}_{j\ell}|}$$

- *Jaccard/Tanimoto distance: For binary vectors ie $\boldsymbol{x}_i \in \{0,1\}^q$*

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{|\boldsymbol{x}_i|^2 + |\boldsymbol{x}_j|^2 - \boldsymbol{x}_i \cdot \boldsymbol{x}_j}$$

$\boldsymbol{x}_i \cdot \boldsymbol{x}_j = \sum_{\ell=1}^{q} \mathrm{x}_{i\ell}\mathrm{x}_{j\ell} = \sum_{\ell=1}^{q} \mathrm{x}_{i\ell} \wedge \mathrm{x}_{j\ell}$ *and* $|\mathbf{x}_i|^2 = \sum_{\ell=1}^{q} \mathrm{x}_{i\ell}^2$

k-Nearest Neighbors (kNN) regression

Some properties of kNN estimators include

- *kNearest Neighbors (kNN) essentially performs regression by averaging the responses of the nearest neighbors of* $\mathbf{x}^*$.
- `kNN` *somewhat performs smoothing (filtering)*
- *The estimated response* $\hat{Y}^*_{\texttt{kNN}}$ *for* $\mathbf{x}^*$ *is estimator of the average response which is the conditional expectation of* Y *given* $\mathbf{x}^*$

$$\hat{Y}^*_{\texttt{kNN}} = \mathbb{E}[\widehat{Y^*|\mathbf{x}^*}]$$

- `kNN` *provides the most basic form of nonparametric regression*
- *Since the fundamental building block of* `kNN` *is the distance measure, one can easily perform regression beyond the traditional setting where the predictors are numeric. eg. Regression vectors of binary) indicator attributes*

$$\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \{0,1\}^p$$

k-Nearest Neighbors (kNN) regression

Limitations of the basic k-Nearest Neighbors approach:

1. *Equidistance: All neighbors are given the same contribution to the estimate of the response;*

$$\hat{Y}^*_{\texttt{kNN}} = \hat{f}_{\texttt{kNN}}(\mathbf{x}^*) = \frac{1}{k}\sum_{\mathbf{x}_j \in \mathcal{V}_k(\mathbf{x}^*)} Y_j = \sum_{\mathbf{x}_j \in \mathcal{V}_k(\mathbf{x}^*)} w_j Y_j$$

   where $w_j = \frac{1}{k} = \mathtt{constant}$ *for all points in* $\mathcal{V}_k(\mathbf{x}^*)$.

2. *No model, no interpretability: There is no underlying model, therefore no interpretation of the response relative to the predictor variables*
3. *Computationally intensive: Predictions are computationally very intensive, due to the fact that for each new observation, the whole dataset must be traversed to compute the response*

Weighted kNearest Neighbors regression

- *Exponential decay:*

$$w_j = \frac{e^{-d_j^*}}{\sum_{l=1}^{k} e^{-d_l^*}}$$

- *Inverse distance:*

$$w_j = \frac{\frac{1}{1+d_j^*}}{\sum_{l=1}^{k} \frac{1}{1+d_l^*}}$$

Note that we define the weights so as to preserve convexity, namely

$$\sum_{j=1}^{k} w_j = 1$$

Question: *How can non convexity affect the weighted kNN method?*

Yet another motivating example for regression

Comparing Criterion based techniques on a toy problem

- Let's consider the quintic function

$$f(\mathrm{x}) = 1 - \frac{1}{2}\mathrm{x} + \mathrm{x}^2 - 2\mathrm{x}^3 + 3\mathrm{x}^5 \qquad \mathrm{x} \in [-1, +1]$$

- Let $n = 400$ be the sample size, and $\sigma^2 = 0.3^2$, and let's generate

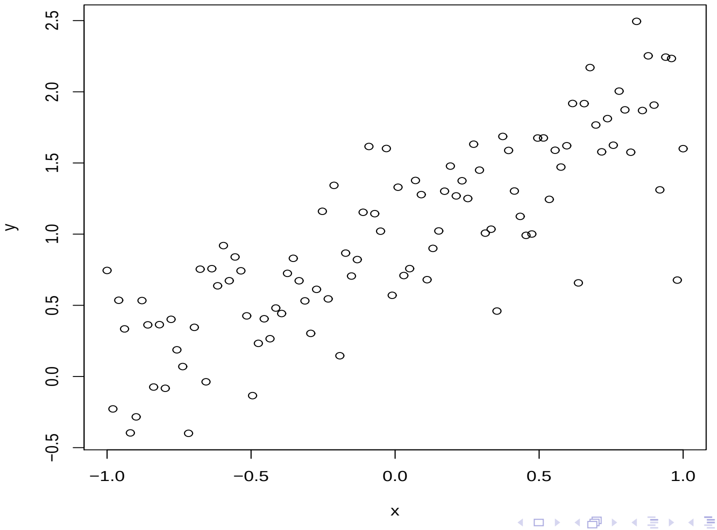
$$y_i = f(\mathrm{x}_i) + \epsilon_i$$

  where $\epsilon_i \sim N(0, \sigma^2)$

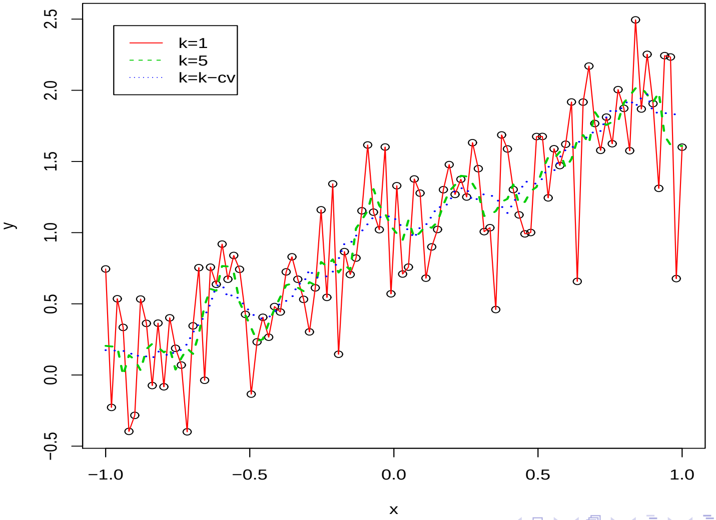
- Let's use kNearest Neighbors regression to try and recover the original function $f(\mathrm{x})$ from the data. For our purposes, we'll use several distances along with several values of k.

Question: How does k Nearest Neighbors Regression fare on this task?

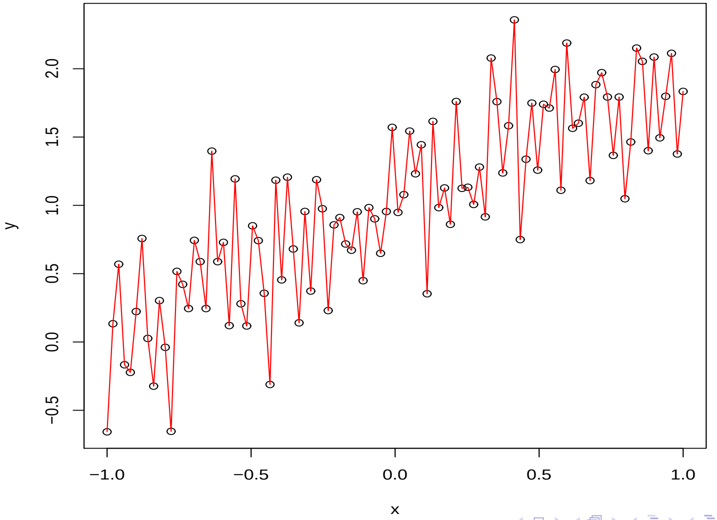
Simple function with large noise



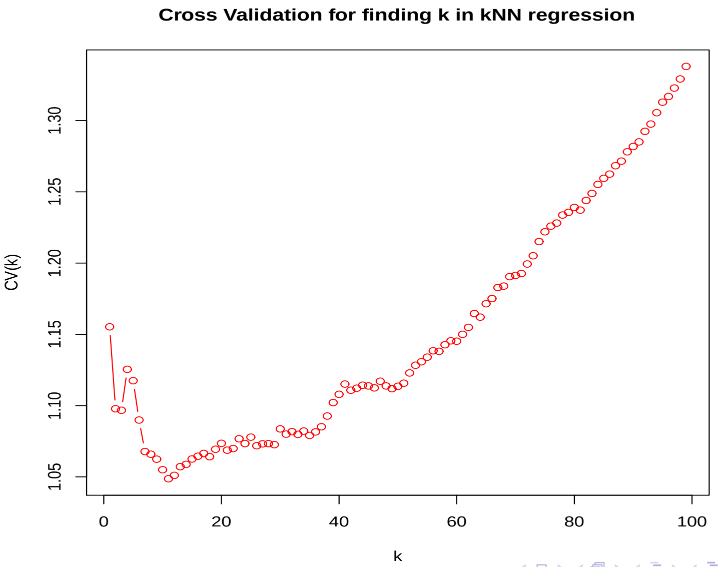
Fitting using kNearest Neighbors



What does Mr Memory do?



Cross Validation on a simple kNN



Extending the kNearest Neighbors Idea

Basic of Nonparametric Regression

Smoothing and Prediction

Cost Functions and Risk Functionals

- *Theorem: Under regularity conditions,*

$$f^*(X) = \mathbb{E}[Y|X] = \operatorname*{arg\,min}_f \mathbb{E}[(Y - f(X))^2]$$

Under the squared error loss, the optimal function f^* that yields the best prediction of Y given X is no other than the expected value of Y given X.

- *Since we know neither $p_{XY}(\mathrm{x}, \mathrm{y})$ nor $p_X(\mathrm{x})$, the conditional expectation*

$$\mathbb{E}[Y|X] = \int_{\mathcal{Y}} \mathrm{y} p_{Y|X}(\mathrm{y})(d\mathrm{y}) = \int_{\mathcal{Y}} \mathrm{y} \frac{p_{XY}(\mathrm{x}, \mathrm{y})}{p_X(\mathrm{x})} d\mathrm{y}$$

cannot be directly computed.

The nonparametric paradigm seeks to estimate $\mathbb{E}[Y|X]$ directly.

Nonparametric Regression

- *The Nadaraya-Watson kernel regression estimator*

$$\hat{f}_{\mathtt{NW}}(\mathrm{x}) = \widehat{\mathbb{E}[Y|\mathrm{x}]} = \frac{\displaystyle\sum_{i=1}^{n} K\left(\frac{\mathrm{x}-X_i}{h}\right) y_i}{\displaystyle\sum_{i=1}^{n} K\left(\frac{\mathrm{x}-X_i}{h}\right)}$$

- $\hat{f}_{\mathtt{NW}}(\mathrm{x})$ *is nonparametric as it directly estimates* f *without a priori imposing a form to* f
- *The bandwidth* h *is often selected by cross validation*
- *Asymptotic unbiasedness of* $\hat{f}_{\mathtt{NW}}(\mathrm{x})$

$$\lim_{n\to\infty} \mathbb{E}(\hat{f}_{\mathtt{NW}}(\mathrm{x})) - f(\mathrm{x}) = 0$$

Nonparametric Regression

- *Weighted average formulation of the Naradaya-Watson*

$$\hat{f}_{\texttt{NW}}(\mathrm{x}) = \sum_{i=1}^{n} w_i Y_i$$

where

$$w_i = \frac{K\left(\dfrac{\mathrm{x} - X_i}{h}\right)}{\displaystyle\sum_{j=1}^{n} K\left(\dfrac{\mathrm{x} - X_j}{h}\right)}$$

- *Clearly, convexity is preserved, i.e.,*

$$\sum_{j=1}^{k} w_j = 1$$

- *The bandwidth is often estimated by cross validation.*

Common kernels in nonparametric regression

- *Gaussian (normal) kernel:*

$$K(\nu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\nu^2} \qquad \nu \in (-\infty, +\infty)$$

- *Epanechnikov kernel:*

$$K(\nu) = \frac{3}{4}(1 - \nu^2) \qquad \nu \in [-1, +1]$$

- *Triangle kernel:*

$$K(\nu) = (1 - |\nu|) \qquad \nu \in [-1, +1]$$

- *Biweight kernel:*

$$K(\nu) = \frac{15}{16}(1 - \nu^2)^2 \qquad \nu \in [-1, +1]$$

Bias-Variance Trade-off in Regression

1. *The Mean Squared Error (MSE)*

$$\text{MSE}\left\{\hat{f}(\mathrm{x}_i)\right\} = \mathbb{E}\left[(\hat{f}(\mathrm{x}_i) - f(\mathrm{x}_i))^2\right]$$

2. *Theoretical Bias-Variance Decomposition in Regression*

$$\text{MSE}\left\{\hat{f}(\mathrm{x}_i)\right\} = \left[\text{Bias}\left\{\hat{f}(\mathrm{x}_i)\right\}\right]^2 + \text{variance}\left\{\hat{f}(\mathrm{x}_i)\right\}$$

3. *The bias is given by*

$$\text{Bias}\left\{\hat{f}(\mathrm{x}_i)\right\} = \mathbb{E}(\hat{f}(\mathrm{x}_i)) - f(\mathrm{x}_i)$$

4. *The variance is given by*

$$\text{variance}\left\{\hat{f}(\mathrm{x}_i)\right\} = \mathbb{E}\left[(\hat{f}(\mathrm{x}_i) - \mathbb{E}(\hat{f}(\mathrm{x}_i)))^2\right]$$

Bias-Variance Trade-off in Regression

1. *Empirical Bias-Variance Decomposition in Regression*
   - *For each estimator $\hat{f}$ of f, we compute the estimates of pointwise bias of $\hat{f}$ by generating many different realizations of $\hat{f}$ from different samples, and then averaging.*

$$\hat{f}_j, \qquad \text{for } j = 1, \cdots, m$$

   - *For each point x_i in the domain, we generate m different samples, and for each sample we form the corresponding $\hat{f}_j$, for $j = 1, \cdots, m$.*

$$\hat{f}_j(\mathbf{x}_i) \quad i = 1, \cdots, n$$

2. *Simulations of this type help understand how bias and variance are decomposed in practice on real life problems*

Bias-Variance Trade-off in Regression

The pointwise bias of $\hat{f}$ at x_i is estimated by

$$\widehat{\text{Bias}}\left\{\hat{f}(\mathrm{x}_i)\right\} = \frac{1}{m}\sum_{j=1}^{m}\hat{f}_j(\mathrm{x}_i) - f(\mathrm{x}_i),$$

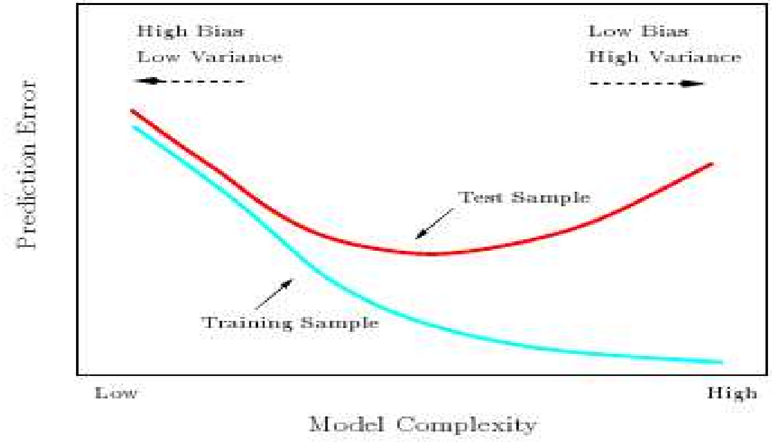
and the pointwise variance of $\hat{f}$ at x_i is estimated by

$$\widehat{\text{variance}}\left\{\hat{f}(\mathrm{x}_i)\right\} = \frac{1}{m-1}\sum_{j=1}^{m}\left(\hat{f}_j(\mathrm{x}_i) - \frac{1}{m}\sum_{\nu=1}^{m}\hat{f}_\nu(\mathrm{x}_i)\right)^2.$$

The estimate of the pointwise mean squared error at x_i is then given by

$$\widehat{\text{MSE}}\left\{\hat{f}(\mathrm{x}_i)\right\} = \left[\widehat{\text{Bias}}\left\{\hat{f}(\mathrm{x}_i)\right\}\right]^2 + \widehat{\text{variance}}\left\{\hat{f}(\mathrm{x}_i)\right\} = \frac{1}{m}\sum_{j=1}^{m}[\hat{f}_j(\mathrm{x}_i) - f(\mathrm{x}_i)$$

Effect of Bias-Variance Dilemma of Prediction



- *Optimal Prediction achieved at the point of bias-variance trade-off.*

Bias-Variance Trade-off in Regression

- *Average Predictive Squared Error (APSE):*

$$APSE(\hat{f}(\mathbf{x}_i)) = PMSE(\hat{f}(\mathbf{x}_i)) = \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}(Y_i - \hat{f}(\mathbf{x}_i))^2\right]$$

- *Cross Validation Squared Error (CVSE):*

$$\mathbb{E}\left[(Y_i - \hat{f}^{(-i)}(\mathbf{x}_i))^2\right] = \sigma^2 + \mathtt{MSE}(\hat{f}(\mathbf{x}_i))$$

- *Interesting formulation of Cross Validation:*

$$\mathtt{CV}(\hat{f}(\mathbf{x}_i)) = \sum_{i=1}^{n}\frac{(Y_i - \hat{f}(\mathbf{x}_i))^2}{n(1-[\mathbf{H}]_{ii})^2}$$

Important result:

$$\mathbb{E}\left[\mathtt{CV}(\hat{f}(\mathbf{x}_i))\right] \approx APSE(\hat{f}(\mathbf{x}_i))$$

The Cross validation score is an asymptotically unbiased estimator of the APSE.

References

James, G, Witten, D, Hastie, T and Tibshirani, R (2013). *An Introduction to Statistical Learning with Applications in R*. Springer, New York, (e-ISBN: 978-1-4614-7138-7),(2013)

Clarke, B, Fokoué, E and Zhang, H (2009). *Principles and Theory for Data Mining and Machine Learning*. Springer Verlag, New York, (ISBN: 978-0-387-98134-5), (2009)

J. J. Faraway(2002). *Practical Regression and ANOVA using* R. Lecture Notes contributed to the R project, (2002)