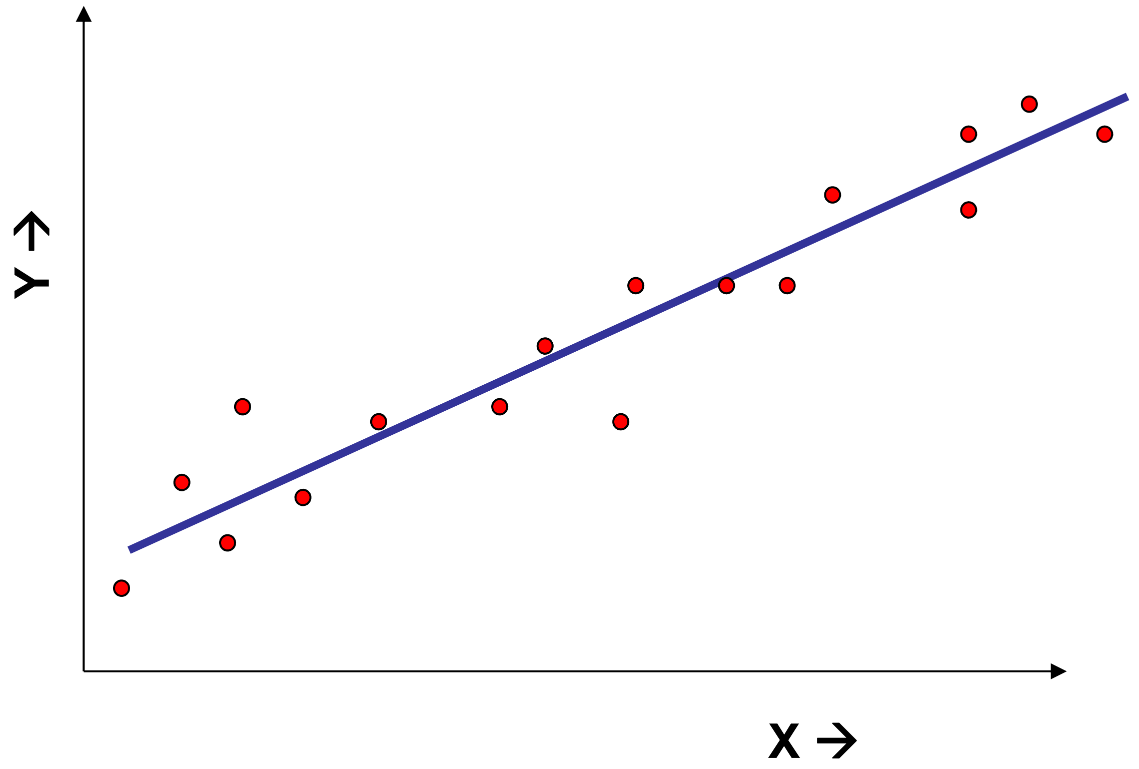
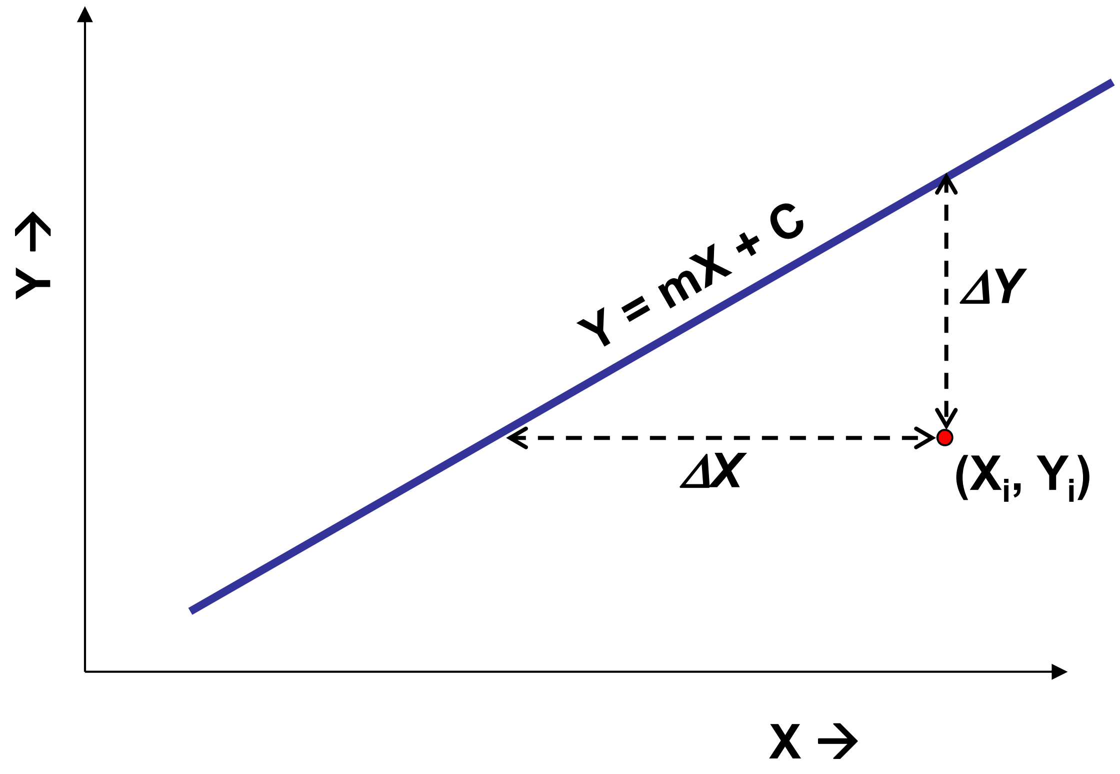
Linear Least Square Regression of a line



A simple and trivial looking problem, but a good illustration



Assume X to be the independent variable with no errors.

Errors are only with Y.

Reduce deviations, $d = \Delta Y$.

METHOD - I

Cost Function:

$$E = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} (Y_i - \tilde{Y}_i)^2$$

$$= \sum_{i=1}^{N} (Y_i - mX_i - C)^2$$

Minimize using derivatives:

$$\frac{\partial E}{\partial m} = -2\sum_{i=1}^{N} (Y_i - mX_i - C)(-X_i) = 0;$$

$$\frac{\partial E}{\partial C} = -\sum_{i=1}^{N} 2(Y_i - mX_i - C) = 0;$$

Re-arranging, we get Normal Equations:

$$NC + \left(\sum_{i=1}^{N} X_i\right)m = \sum_{i=1}^{N} Y_i;$$

$$\left(\sum_{i=1}^{N} X_i\right)C + \left(\sum_{i=1}^{N} X_i^2\right)m = \sum_{i=1}^{N} (X_i Y_i);$$

Solving, we get:

$$m = \frac{N\sum_{i=1}^{N}(X_iY_i) - \sum_{i=1}^{N}X_i\sum_{i=1}^{N}Y_i}{DEN};$$

$$C = \frac{\sum_{i=1}^{N}Y_i\sum_{i=1}^{N}X_i^2 - \sum_{i=1}^{N}X_i\sum_{i=1}^{N}(X_iY_i)}{DEN};$$

where,

$$DEN = N\sum_{i=1}^{N}X_i^2 - (\sum_{i=1}^{N}X_i)^2$$

In parametric form:

$$m = \frac{N^2\sigma_{XY} - N^2\mu_X\mu_Y}{N^2\sigma_X^2 - N^2\mu_X^2} = \frac{\sigma_{XY} - \mu_X\mu_Y}{\sigma_X^2 - \mu_X^2}$$

In parametric form:

$$m = \frac{N^2\sigma_{XY} - N^2\mu_X\mu_Y}{N^2\sigma_X^2 - N^2\mu_X^2} = \frac{\sigma_{XY} - \mu_X\mu_Y}{\sigma_X^2 - \mu_X^2}$$

$$C = \frac{\sum_{i=1}^{N} Y_i \sum_{i=1}^{N} X_i^2 - \sum_{i=1}^{N} X_i \sum_{i=1}^{N} (X_i Y_i)}{N\sum_{i=1}^{N} X_i^2 - (\sum_{i=1}^{N} X_i)^2};$$

$$= \frac{N^2\mu_Y\sigma_X^2 - N^2\mu_X\sigma_{XY}}{N^2\sigma_X^2 - N^2\mu_X^2} = \frac{\mu_Y\sigma_X^2 - \mu_X\sigma_{XY}}{\sigma_X^2 - \mu_X^2}$$

Check from above that the LSQ-line passes through the point: (μ_X, μ_Y). Thus shift the origin to the point: (μ_X, μ_Y).

The equation of the line in the transformed space:

$$m' = \sigma_{XY} \big/ \sigma_X^2 \; ; C' = 0.$$

METHOD - II

Solving the same, using matrix concepts:

$$Y = mX + C \quad \Rightarrow \mathrm{mX} + \mathrm{C} = \mathrm{Y};$$

$$\begin{bmatrix} \mathrm{X} & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = Y;$$

Any two points on the line, can give us the parameters:

$$\begin{bmatrix} \mathrm{X}_1 & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = Y_1; \; \begin{bmatrix} \mathrm{X}_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = Y_2;$$

$$\begin{bmatrix} \mathrm{X}_1 & 1 \\ \mathrm{X}_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix};$$

Thus:

$$\begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \left(\frac{1}{X_1 - X_2}\right) \begin{bmatrix} 1 & -1 \\ -X_2 & X_1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

If you use this: $$\begin{bmatrix} X_1 & 1 \end{bmatrix}\begin{bmatrix} m \\ C \end{bmatrix} = Y_1; \quad \begin{bmatrix} X_2 & 1 \end{bmatrix}\begin{bmatrix} m \\ C \end{bmatrix} = Y_2;$$

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix}\begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix};$$

${}^{N}C_2$ lines may be obtained for each pair.

In case of best fit:

We are basically trying to solve an ill-posed problem, where:

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ & \\ & \\ X_N & 1 \end{bmatrix}\begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ \\ Y_N \end{bmatrix};$$

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ & \\ & \\ X_N & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ \\ Y_N \end{bmatrix};$$

Take this as: Aβ = B,
where A is a non-square
(or even singular square) matrix.

Use Pseudo-inverse in this case:

$$A\beta = B \Rightarrow A^T A\beta = A^T B;$$

$$(A^T A)\beta = A^T B \qquad \Rightarrow \qquad \beta = (A^T A)^{-1} A^T B;$$

$$\beta = A^+ B;$$

$$where, \quad A^+ = (A^T A)^{-1} A^T; \text{ is the Pseudo-inverse.}$$

(A^TA) is square and assumed to be non-singular (generally). If not, look for alternative formula (hang on, for now)

A^+A or AA^+ is not equal to I (except non-singular square A), but I_p.

$$A = \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ & \\ X_N & 1 \end{bmatrix};$$

$$A\beta = B \Rightarrow A^T A\beta = A^T B;$$
$$\beta = (A^T A)^{-1} A^T B;$$

$$B = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ Y_N \end{bmatrix};$$

$$A^T A = \begin{bmatrix} X_1 & X_2 & & X_N \\ 1 & 1 & & 1 \end{bmatrix} \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ & \\ X_N & 1 \end{bmatrix};$$

$$A^T A = \begin{bmatrix} \sum_{i=1}^{N} X_i^2 & \sum_{i=1}^{N} X_i \\ \sum_{i=1}^{N} X_i & N \end{bmatrix} = N \begin{bmatrix} \sigma_x^2 & \mu_x \\ \mu_x & 1 \end{bmatrix}$$

$$\beta = [\mathrm{m}\ \mathrm{C}]^{\mathrm{T}}$$

$$A^T A = N\begin{bmatrix} \sigma_x^2 & \mu_x \\ \mu_x & 1 \end{bmatrix}; \; (A^T A)^{-1} = (\frac{1}{N.D})\begin{bmatrix} 1 & -\mu_x \\ -\mu_x & \sigma_x^2 \end{bmatrix}$$

where, D= $\sigma_x^2 - \mu_x^2$

$$A^T B = \begin{bmatrix} X_1 & X_2 & & X_N \\ 1 & 1 & & 1 \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \\ Y_N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} X_i Y_i \\ \sum_{i=1}^{N} Y_i \end{bmatrix} = N\begin{bmatrix} \sigma_{XY} \\ \mu_Y \end{bmatrix};$$

Thus:

$$\beta = A^+ B = (A^T A)^{-1} A^T B = \left(1/D\right)\begin{bmatrix} 1 & -\mu_X \\ -\mu_X & \sigma_x^2 \end{bmatrix}\begin{bmatrix} \sigma_{XY} \\ \mu_Y \end{bmatrix}$$

$$= \left(1/D\right)\begin{bmatrix} \sigma_{XY} - \mu_X \mu_Y \\ \sigma_x^2 \mu_Y - \sigma_{XY}\mu_X \end{bmatrix}$$

The solution is same as in LSQ-FIT.

Pseudo-inverse satisfies the Least-square criteria.

Notations in ESLT – Hastie:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Np} \end{bmatrix} = [x_1 \cdots x_N]^T = [\boldsymbol{X}_1 \cdots \boldsymbol{X}_p]$$

i-th Observation (row-vector):

$$x_i^T = [x_{i1} \cdots x_{ip}]; i = 1, \dots, N$$

All observations of j-th variable (e.g. sensor):

$$\boldsymbol{X}_j = [x_{1j} \cdots x_{Nj}]^T; j = 1, \dots, p$$

$$\begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ & \\ 1 & X_N \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ Y_N \end{bmatrix}; p = 1.$$

$$\begin{bmatrix} 1 & X_1^1 & X_1^2 & X_1^3 \\ 1 & X_2^1 & X_2^2 & X_2^3 \\ & . & . & \\ & . & . & \\ 1 & X_N^1 & X_N^2 & X_N^3 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ \\ Y_N \end{bmatrix};$$

$$p = 3$$

$$\begin{bmatrix} 1 & X_1^1 & X_1^2 \\ 1 & X_2^1 & X_2^2 \\ & & \\ 1 & X_N^1 & X_N^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ Y_N \end{bmatrix}; p = 2.$$

$$\begin{bmatrix} 1 & \mathbf{X}_1 & .. & \mathbf{X}_p \end{bmatrix} \begin{bmatrix} \beta_0 \\ . \\ . \\ \beta_p \end{bmatrix} = [\mathbf{Y}]$$

$$(A^T A)X = A^T B$$

$$\Rightarrow \quad \beta = (A^T A)^{-1} A^T B;$$

$$\beta = A^+ B;$$

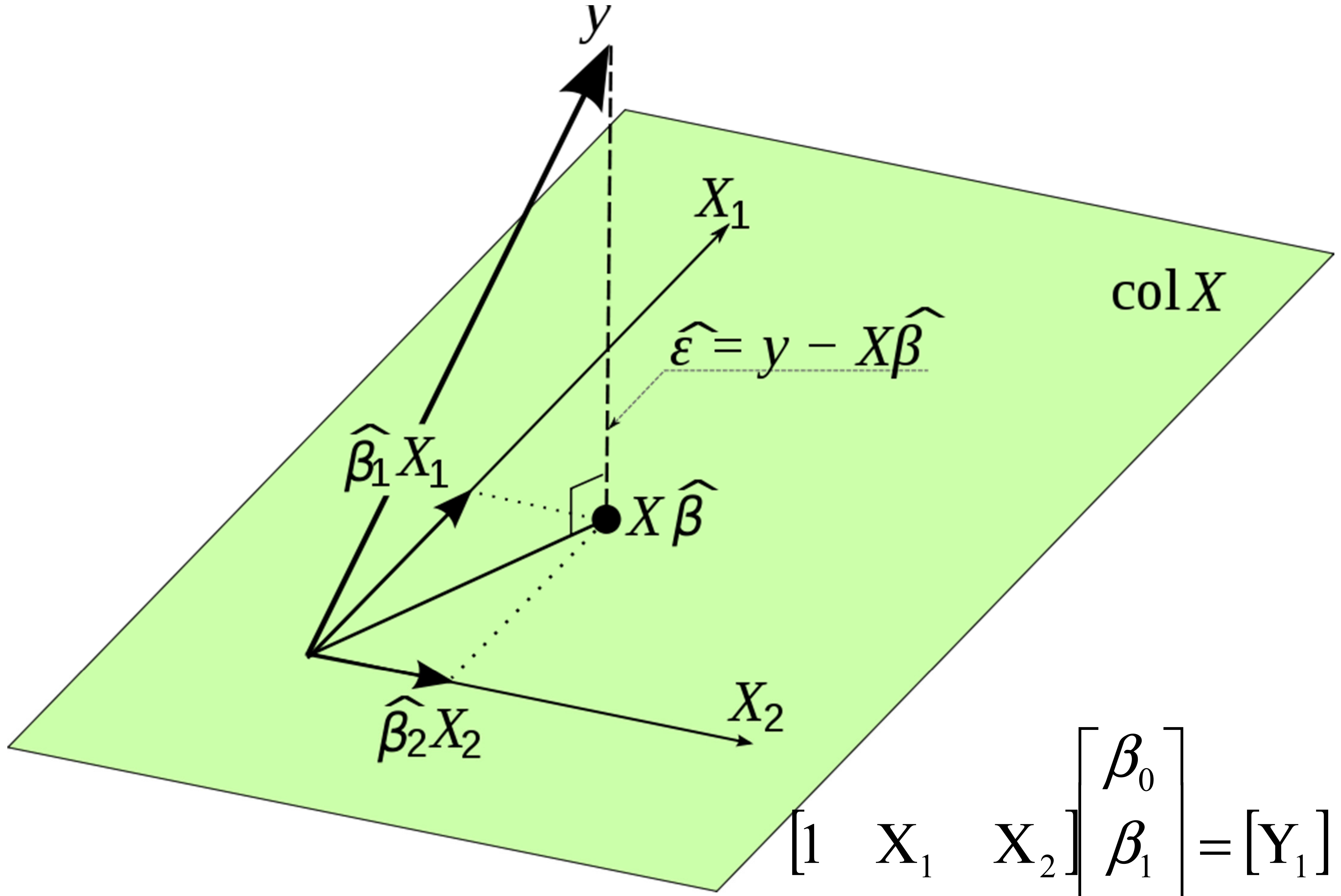
$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ & \\ X_N & 1 \end{bmatrix} \begin{bmatrix} m \\ C \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \\ Y_N \end{bmatrix};$$

$$where, \quad A^+ = (A^T A)^{-1} A^T; \text{ is the Pseudo - inverse.}$$

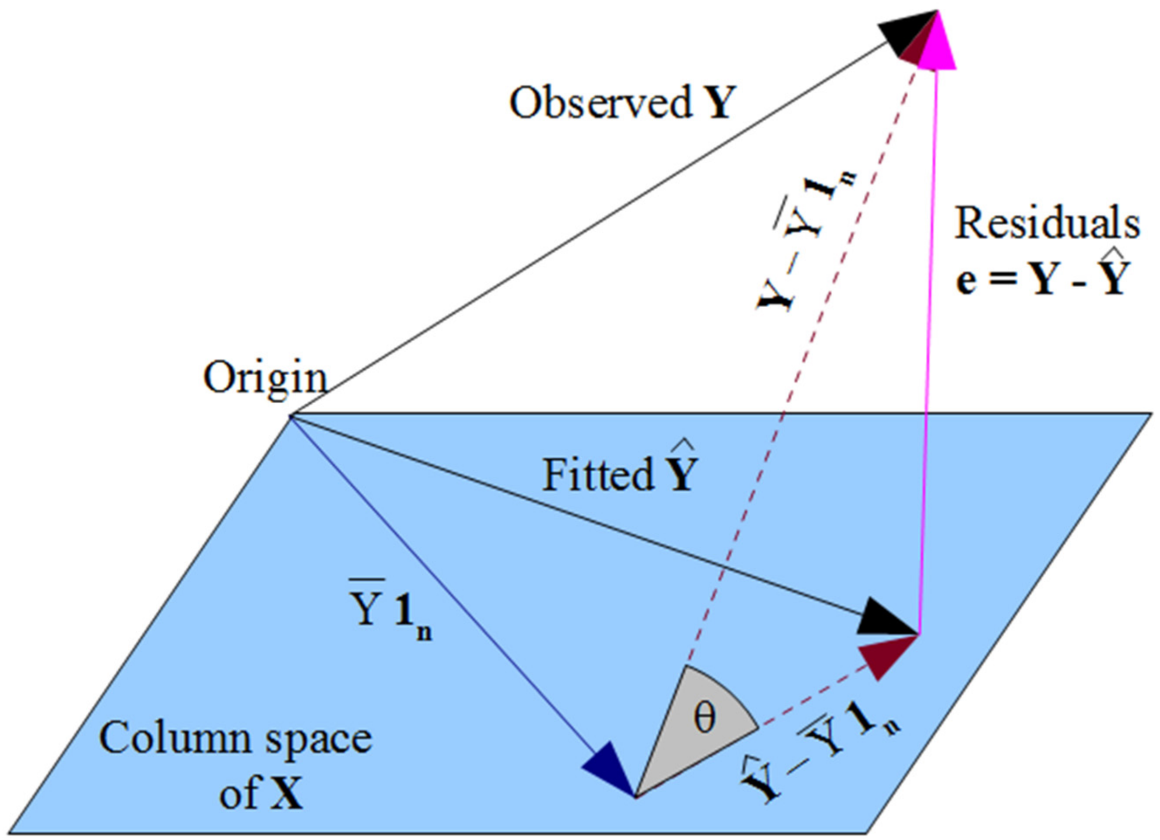
$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

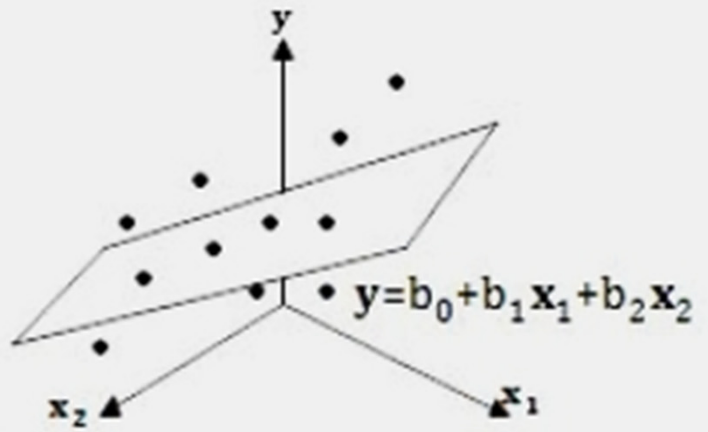
where $\hat{y}_i = \hat{f}(x_i)$. The matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ appearing in equation (3.7) is sometimes called the "hat" matrix because it puts the hat on $\mathbf{y}$.

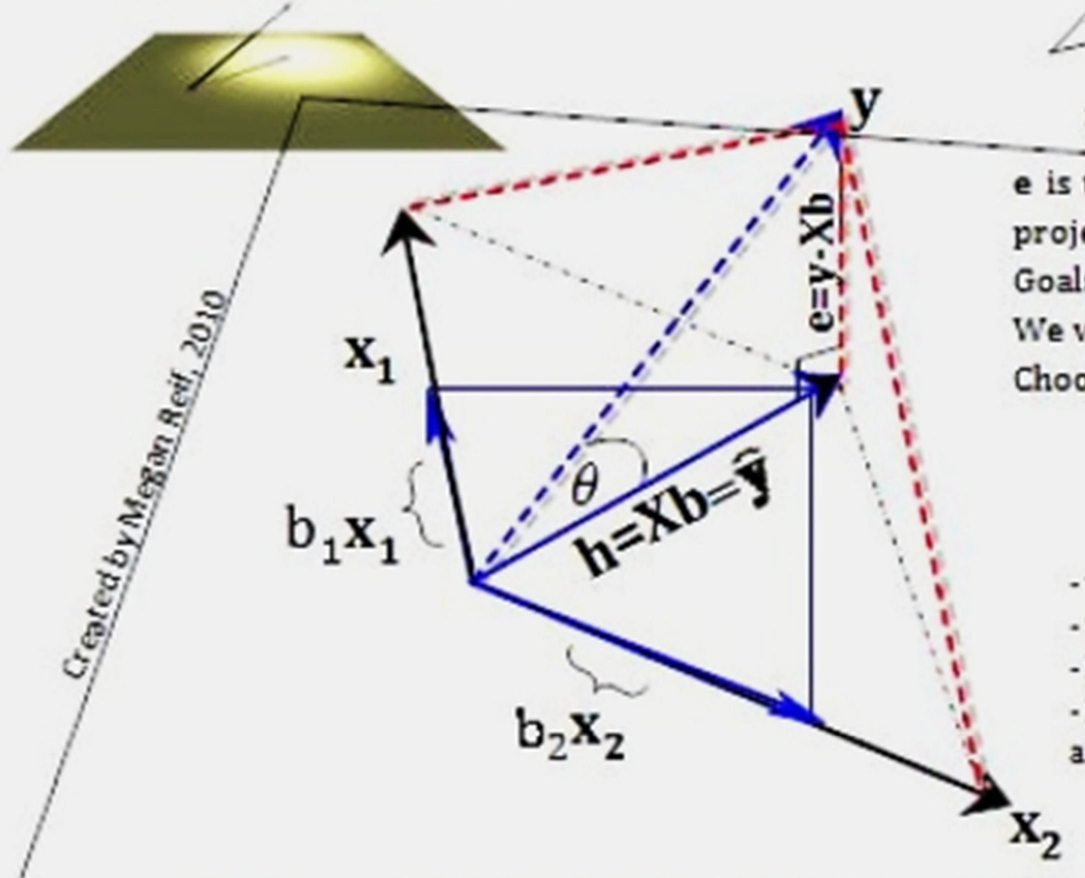


$$\begin{bmatrix} 1 & X_1 & X_2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} Y_1 \end{bmatrix}$$


Observed $\mathbf{Y}$
$\mathbf{Y} - \overline{Y}\,\mathbf{1}_n$
Residuals
$\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$
Origin
Fitted $\hat{\mathbf{Y}}$
$\overline{Y}\,\mathbf{1}_n$
θ
$\hat{\mathbf{Y}} - \overline{Y}\,\mathbf{1}_n$
Column space of $\mathbf{X}$

The vector of observed **y** does not lie in the plane **X**.
OLS swaps out **y** for a vector that IS IN THE PLANE,
but as close as possible to **y**. This vector is selected
through "projection", not unlike shining a light from above
y onto the plane. This calculation is achieved through the
matrix **H, or fitted-value** maker.





e is the "error" between observed vector *y* and
projected vector **h** that we plan to swap for **y**.
Goal: choose **h** to make **e** as small as possible.
We want to minimize the error between **h** and **y**.
Choose **h** perpendicular to error vector **e**.

- when fit is good, angle is small, **h** is long
- when fit is poor, angle large, and **h** is short.
- leverage points will have HIGH FIT:
- leverage points mean that, given
an extreme **X**, **fitted y** will predict observed y

Column space of **X** (flat plane) (our mathematical model):
Columns of **X** are vectors $\mathbf{x}_1, \mathbf{x}_2$ (in 3-space) .. $\mathbf{x}_i ....\mathbf{x}_k$ (in n-space)
(The plane represents all *possible* linear combinations of the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.)

$$\beta = (X^T X)^{-1} X^T y;$$

$$\mathrm{var}(\beta) = E(\beta\beta^T) = E[(X^T X)^{-1} X^T y)(X^T X)^{-1} X^T y)^T]$$

$$= (X^T X)^{-1} X^T E(yy^T) X (X^T X)^{-1}$$

$$= (X^T X)^{-1} \sigma^2 \quad \text{or} \quad \sigma^2 (X^T X)^{-1}$$

Now see chap. 3 of Hastie – ESL book,
for "HAT matrix" in regression;

And come back for related concepts.

In statistics and machine learning, the bias–variance tradeoff (or dilemma) is the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set:

The bias is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The variance is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting: modeling the random noise in the training data, rather than the intended outputs.

The bias–variance decomposition is a way of analyzing a learning algorithm's expected generalization error with respect to a particular problem as a sum of three terms, the bias, variance, and a quantity called the irreducible error, resulting from noise in the problem itself.

This tradeoff applies to all forms of supervised learning: classification, regression (function fitting), and structured output learning. It has also been invoked to explain the effectiveness of heuristics in human learning. <Src: WIKI>

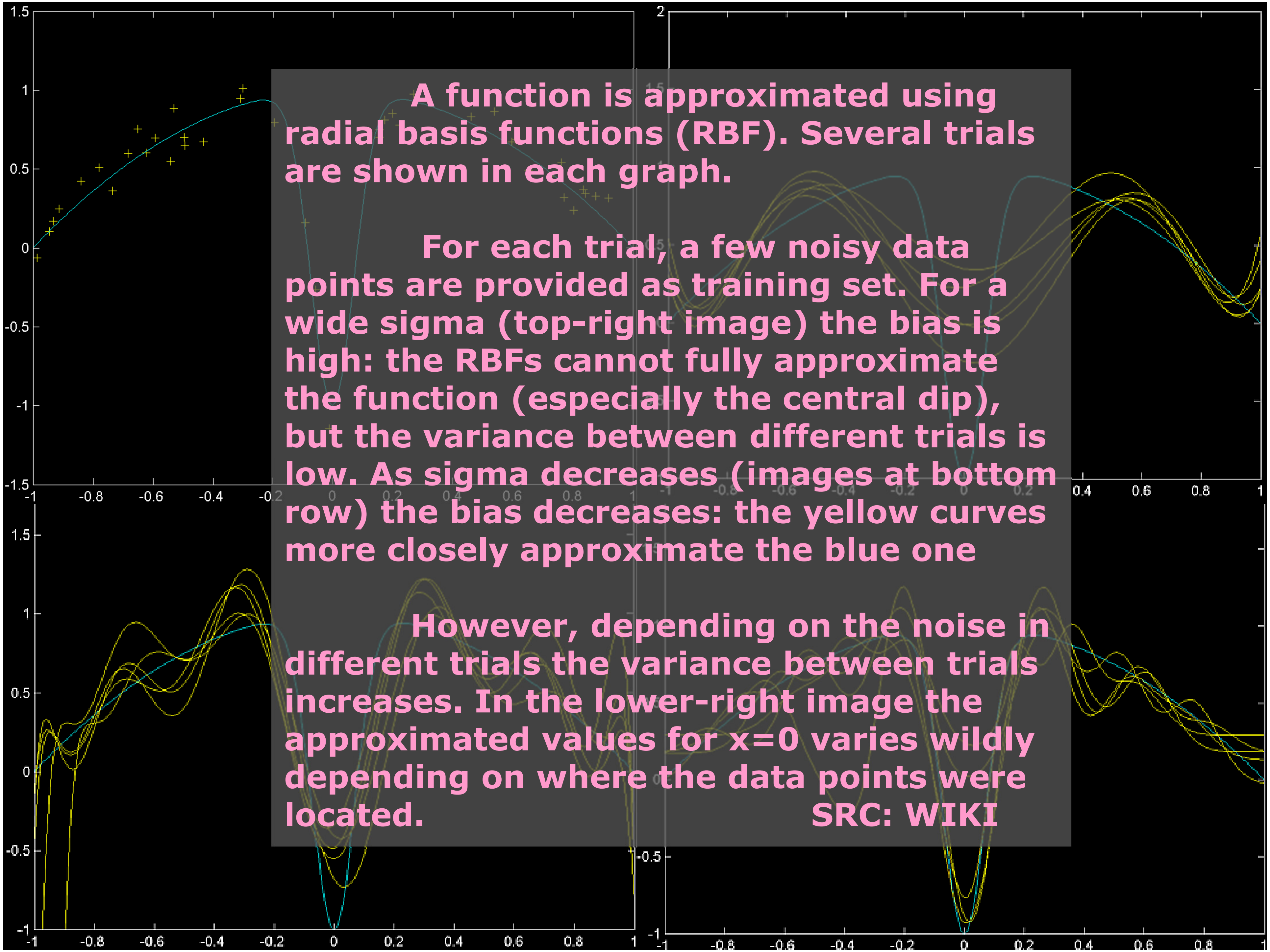
Motivation (WIKI):

The bias–variance tradeoff is a central problem in supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously.

High-variance learning methods may be able to represent their training set well, but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with high bias typically produce simpler models that don't tend to overfit, but may *underfit* their training data, failing to capture important regularities.

Models with low bias are usually more complex (e.g. higher-order regression polynomials), enabling them to represent the training set more accurately. In the process, however, they may also represent a large noise component in the training set, making their predictions less accurate - despite their added complexity.

In contrast, models with higher bias (low variance) tend to be relatively simple (low-order or even linear regression polynomials), but may produce lower variance predictions when applied beyond the training set.



A function is approximated using radial basis functions (RBF). Several trials are shown in each graph.

For each trial, a few noisy data points are provided as training set. For a wide sigma (top-right image) the bias is high: the RBFs cannot fully approximate the function (especially the central dip), but the variance between different trials is low. As sigma decreases (images at bottom row) the bias decreases: the yellow curves more closely approximate the blue one

However, depending on the noise in different trials the variance between trials increases. In the lower-right image the approximated values for x=0 varies wildly depending on where the data points were located. SRC: WIKI

1.2
1.0
High Bias
Low Variance
Low Bias
High Variance

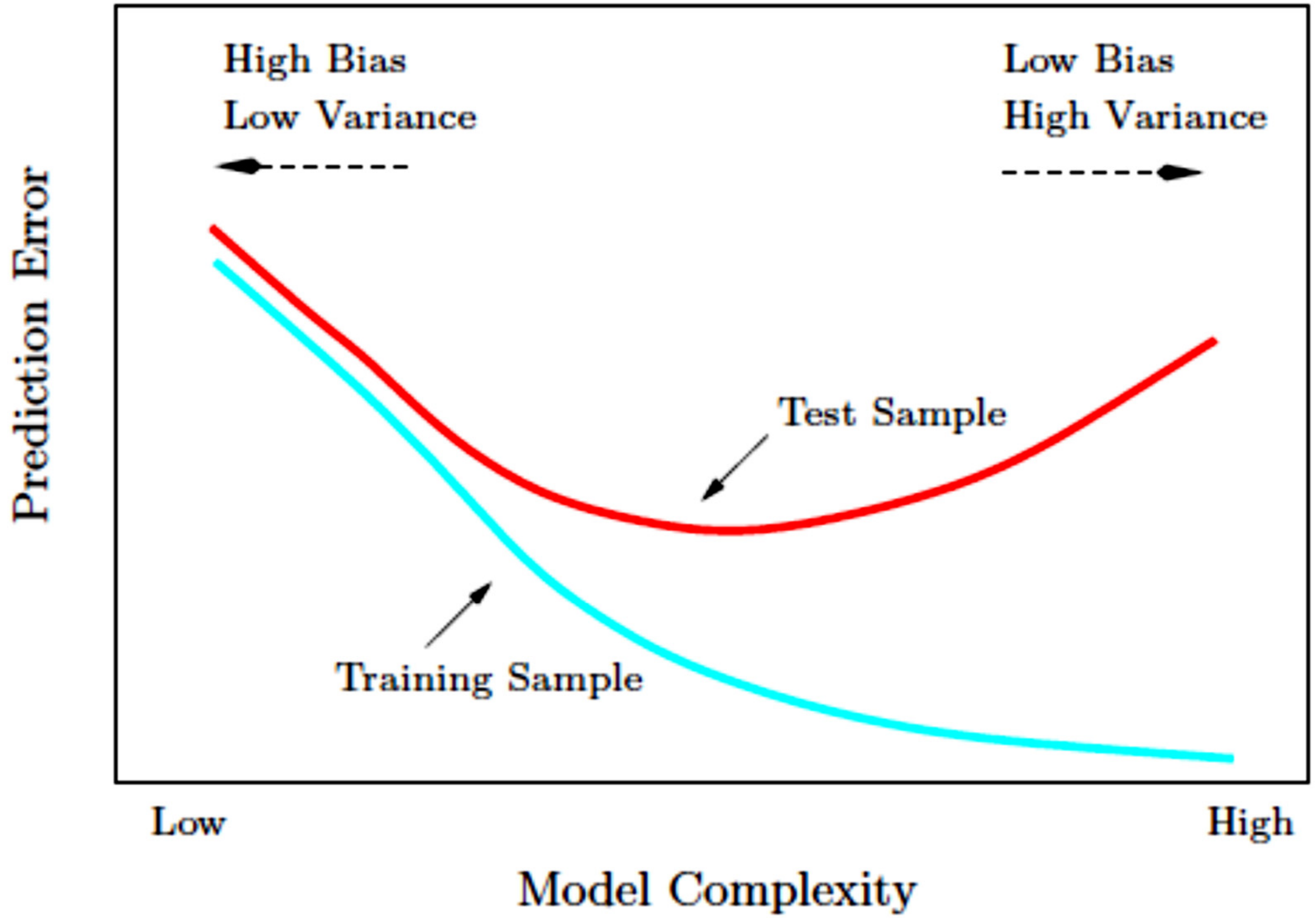
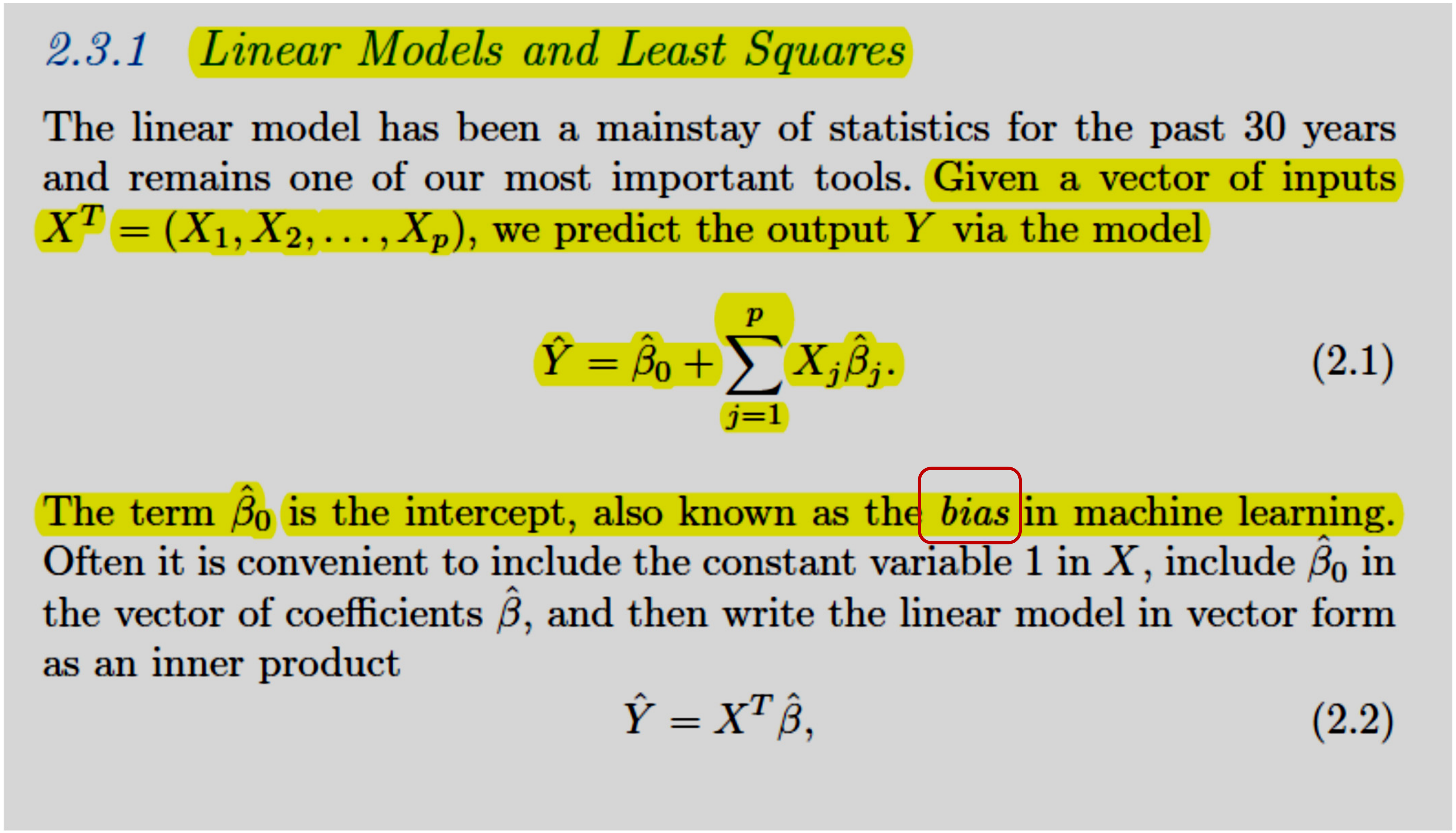


FIGURE 2.11. *Test and training error as a function of model complexity.*



2.3.1 *Linear Models and Least Squares*

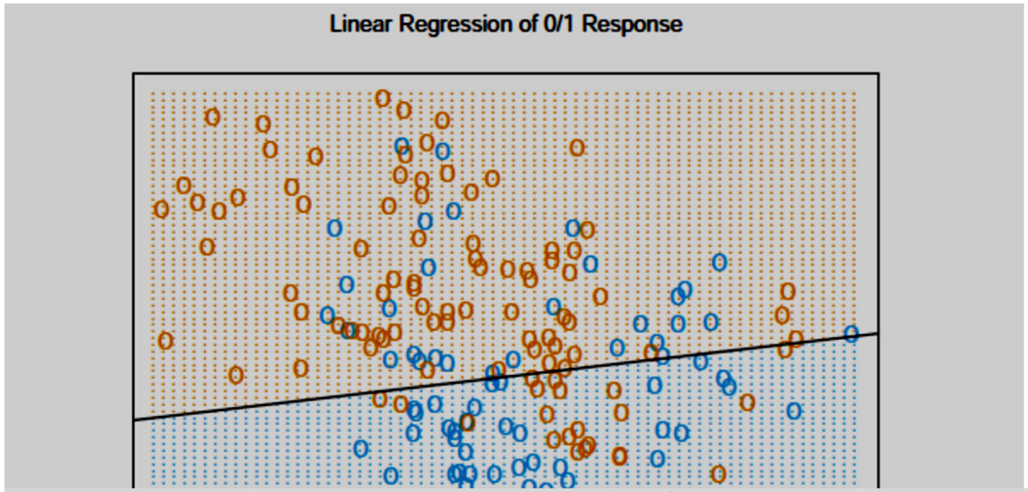
The linear model has been a mainstay of statistics for the past 30 years and remains one of our most important tools. Given a vector of inputs $X^T = (X_1, X_2, \ldots, X_p)$, we predict the output Y via the model

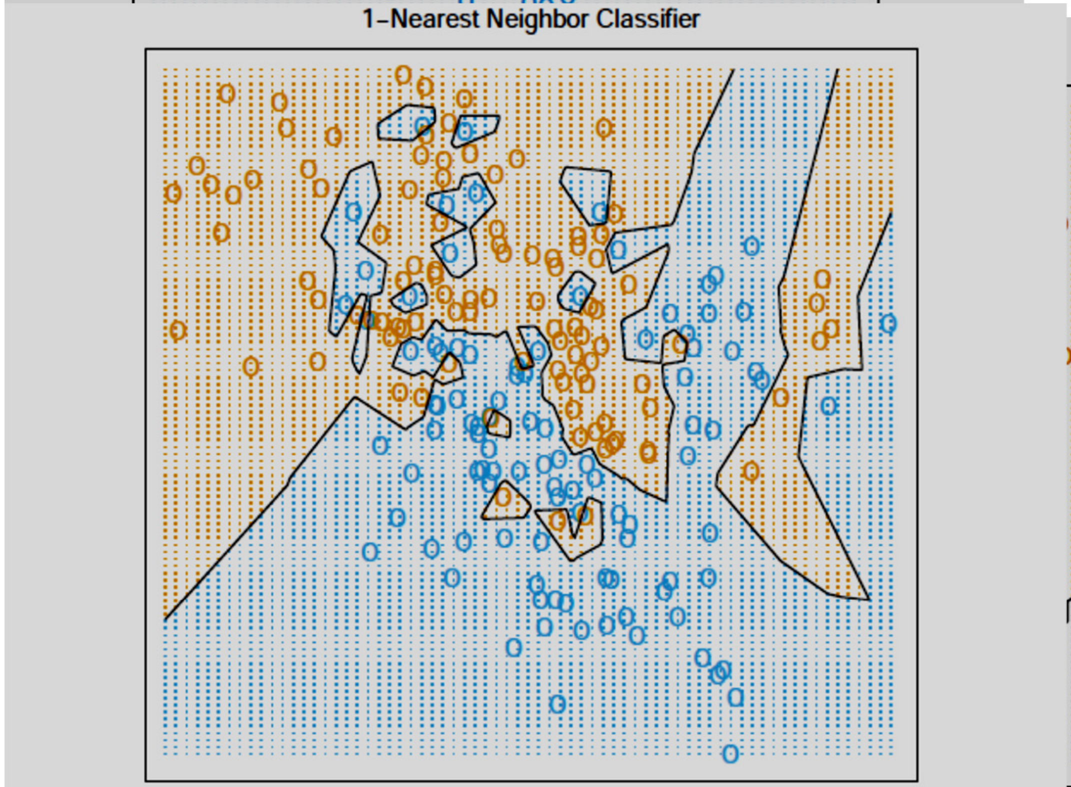
$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j. \tag{2.1}$$

The term $\hat{\beta}_0$ is the intercept, also known as the *bias* in machine learning. Often it is convenient to include the constant variable 1 in X, include $\hat{\beta}_0$ in the vector of coefficients $\hat{\beta}$, and then write the linear model in vector form as an inner product

$$\hat{Y} = X^T \hat{\beta}, \tag{2.2}$$

Lets examine two learning techniques for prediction: the stable but biased linear model and the less stable but apparently less biased class of k-nearest-neighbor estimates.





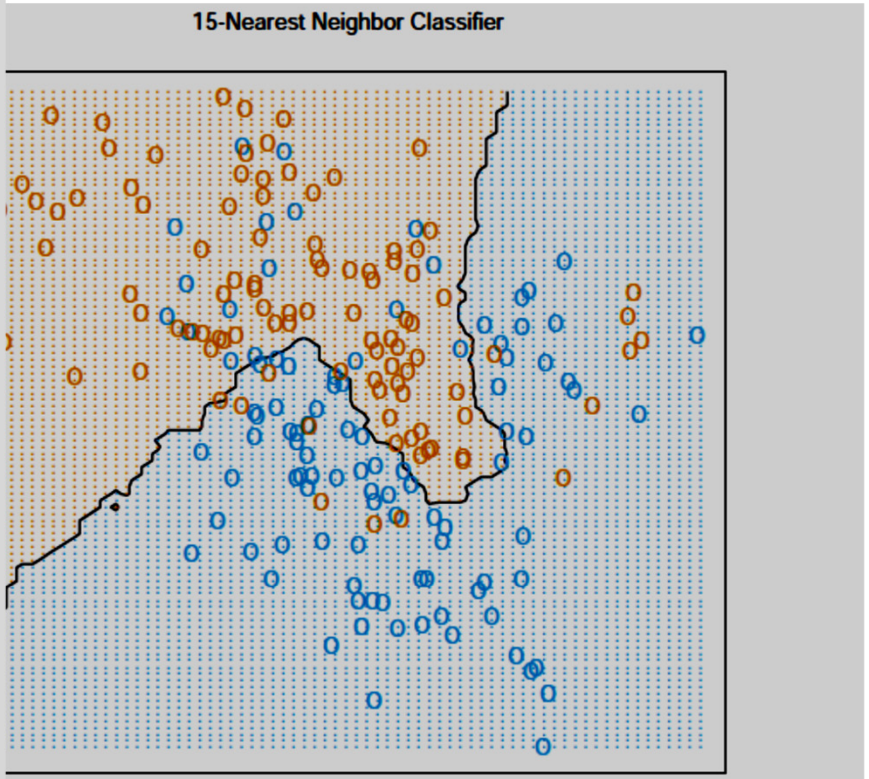


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE = 0, ORANGE = 1)*, and then predicted by 1-nearest-neighbor classification.*

.2. The same classification example in two dimensions as in Fig- classes are coded as a binary variable (BLUE = 0, ORANGE = 1) *and -nearest-neighbor averaging as in (2.8). The predicted class is hence ajority vote amongst the 15-nearest neighbors.*

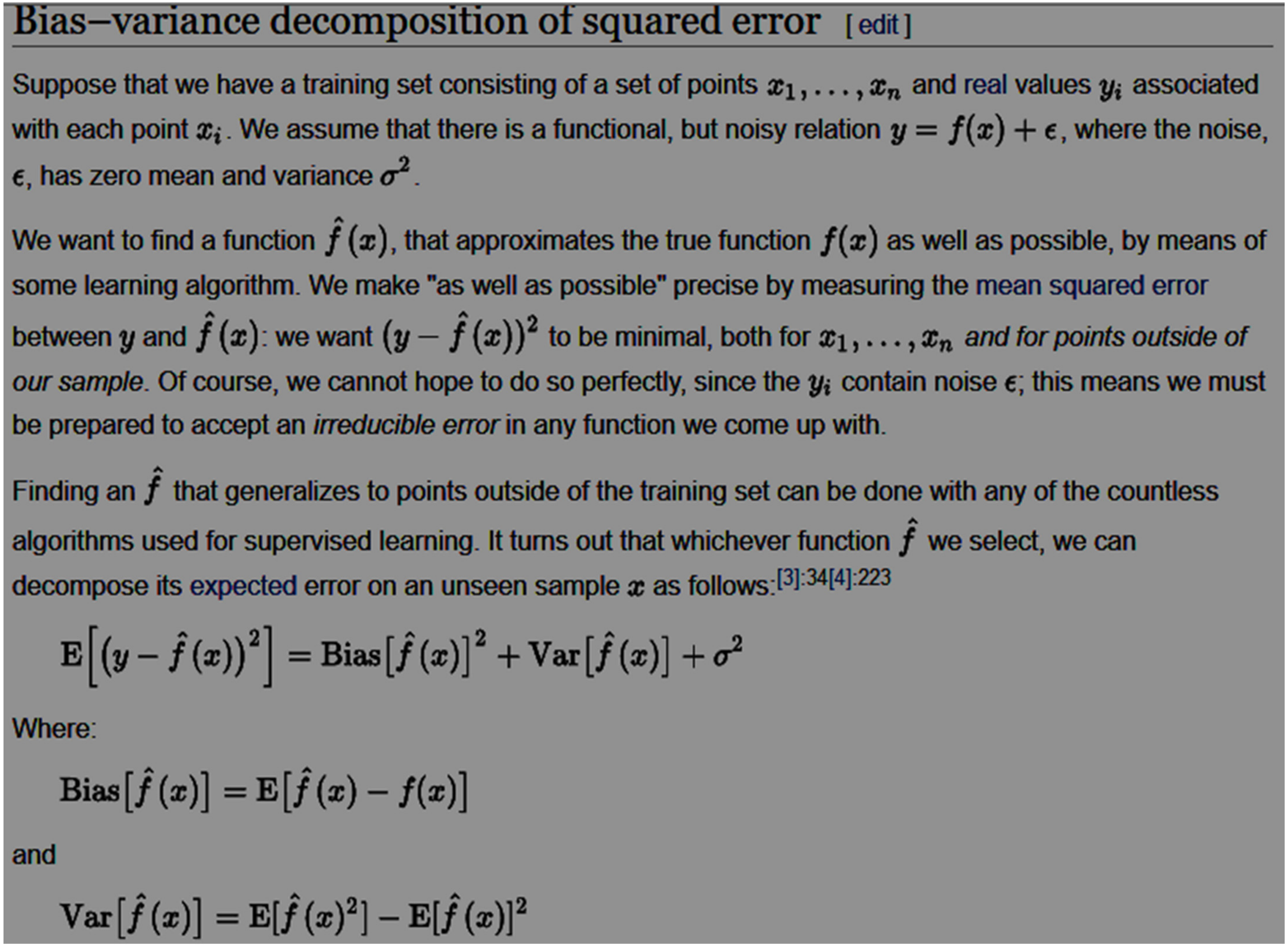
The linear decision boundary from least squares is very smooth, and apparently stable to fit. It does appear to rely heavily on the assumption that a linear decision boundary is appropriate. It **has low variance and potentially high bias.**

On the other hand, the k-nearest-neighbor procedures do not appear to rely on any stringent assumptions about the underlying data, and can adapt to any situation. However, any particular sub-region of the decision boundary depends on a handful of input points and their particular positions, and is **thus wiggly and unstable—high variance and low bias**.

The MSE is generally broken into two components that will become familiar as we proceed: variance and squared bias. Such a decomposition is always possible and often useful, and is known as the **bias–variance decomposition.**

$$\begin{aligned}
\mathrm{MSE}(x_0) &= \mathrm{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\
&= \mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2 + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0).
\end{aligned}$$

REFER : ELT - HASTIE BOOK PP (37, 239, 242)



Bias–variance decomposition of squared error [edit]

Suppose that we have a training set consisting of a set of points $x_1, \ldots, x_n$ and real values y_i associated with each point x_i. We assume that there is a functional, but noisy relation $y = f(x) + \epsilon$, where the noise, ϵ, has zero mean and variance σ^2.

We want to find a function $\hat{f}(x)$, that approximates the true function $f(x)$ as well as possible, by means of some learning algorithm. We make "as well as possible" precise by measuring the mean squared error between y and $\hat{f}(x)$: we want $(y - \hat{f}(x))^2$ to be minimal, both for $x_1, \ldots, x_n$ *and for points outside of our sample*. Of course, we cannot hope to do so perfectly, since the y_i contain noise ϵ; this means we must be prepared to accept an *irreducible error* in any function we come up with.

Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:[3]:34[4]:223

$$\mathrm{E}\Big[\big(y - \hat{f}(x)\big)^2\Big] = \mathrm{Bias}\big[\hat{f}(x)\big]^2 + \mathrm{Var}\big[\hat{f}(x)\big] + \sigma^2$$

Where:

$$\mathrm{Bias}\big[\hat{f}(x)\big] = \mathrm{E}\big[\hat{f}(x) - f(x)\big]$$

and

$$\mathrm{Var}\big[\hat{f}(x)\big] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2$$

$$\mathrm{E}\Big[\big(y-\hat{f}(x)\big)^2\Big] = \mathrm{Bias}\big[\hat{f}(x)\big]^2 + \mathrm{Var}\big[\hat{f}(x)\big] + \sigma^2$$

Where:

$$\mathrm{Bias}\big[\hat{f}(x)\big] = \mathrm{E}\big[\hat{f}(x) - f(x)\big]$$

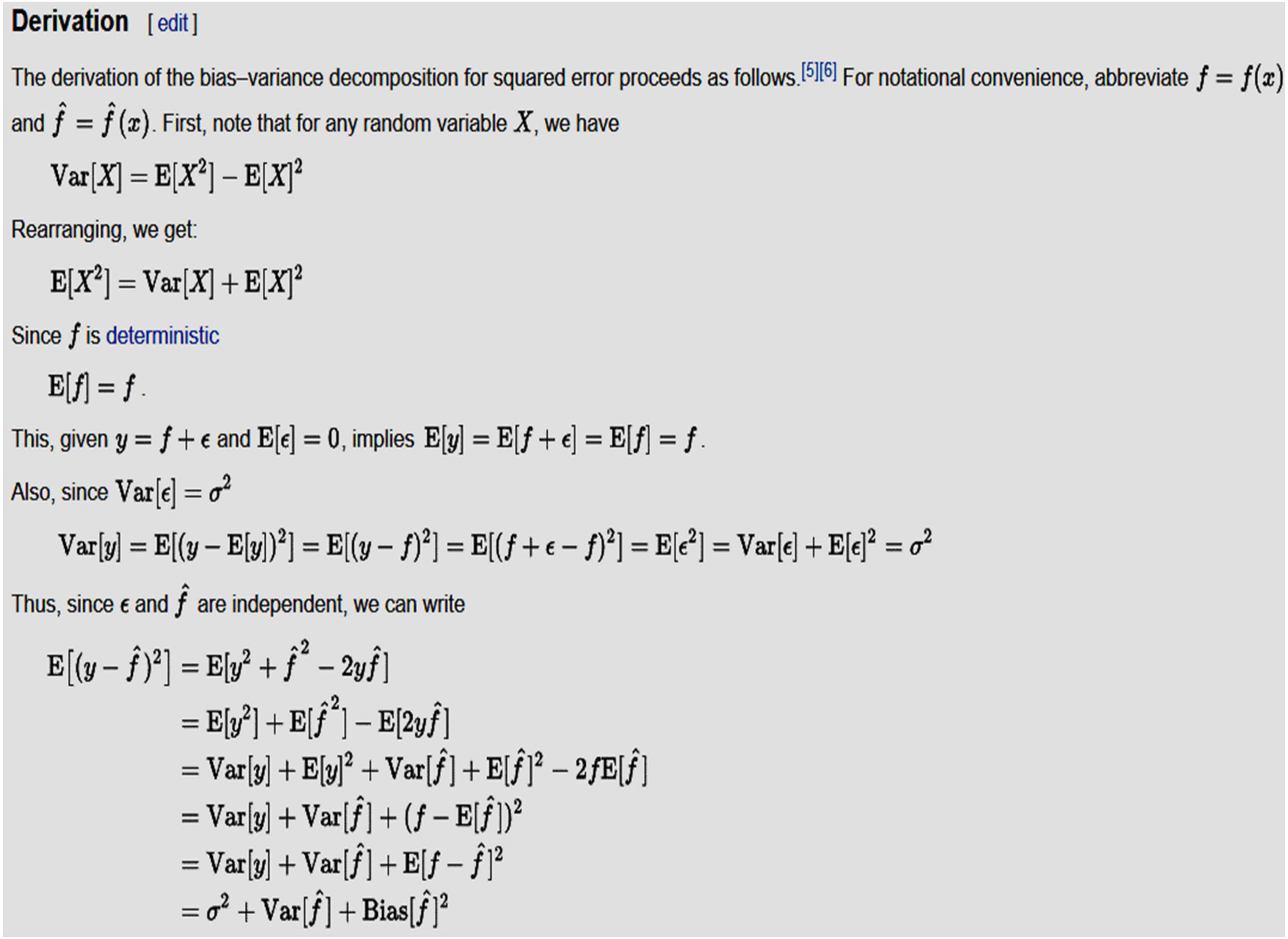
and

$$\mathrm{Var}\big[\hat{f}(x)\big] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2$$

The expectation ranges over different choices of the training set $x_1, \ldots, x_n, y_1, \ldots, y_n$, all sampled from the same (conditional) distribution. The three terms represent:

- the square of the *bias* of the learning method, which can be thought of the error caused by the simplifying assumptions built into the method. E.g., when approximating a non-linear function $f(x)$ using a learning method for linear models, there will be error in the estimates $\hat{f}(x)$ due to this assumption;
- the *variance* of the learning method, or, intuitively, how much the learning method $\hat{f}(x)$ will move around its mean;
- the irreducible error σ^2. Since all three terms are non-negative, this forms a lower bound on the expected error on unseen samples.[3]:34

The more complex the model $\hat{f}(x)$ is, the more data points it will capture, and the lower the bias will be. However, complexity will make the model "move" more to capture the data points, and hence its variance will be larger.



Derivation [edit]

The derivation of the bias–variance decomposition for squared error proceeds as follows.[5][6] For notational convenience, abbreviate $f = f(x)$ and $\hat{f} = \hat{f}(x)$. First, note that for any random variable X, we have

$$\mathrm{Var}[X] = \mathrm{E}[X^2] - \mathrm{E}[X]^2$$

Rearranging, we get:

$$\mathrm{E}[X^2] = \mathrm{Var}[X] + \mathrm{E}[X]^2$$

Since f is deterministic

$$\mathrm{E}[f] = f\,.$$

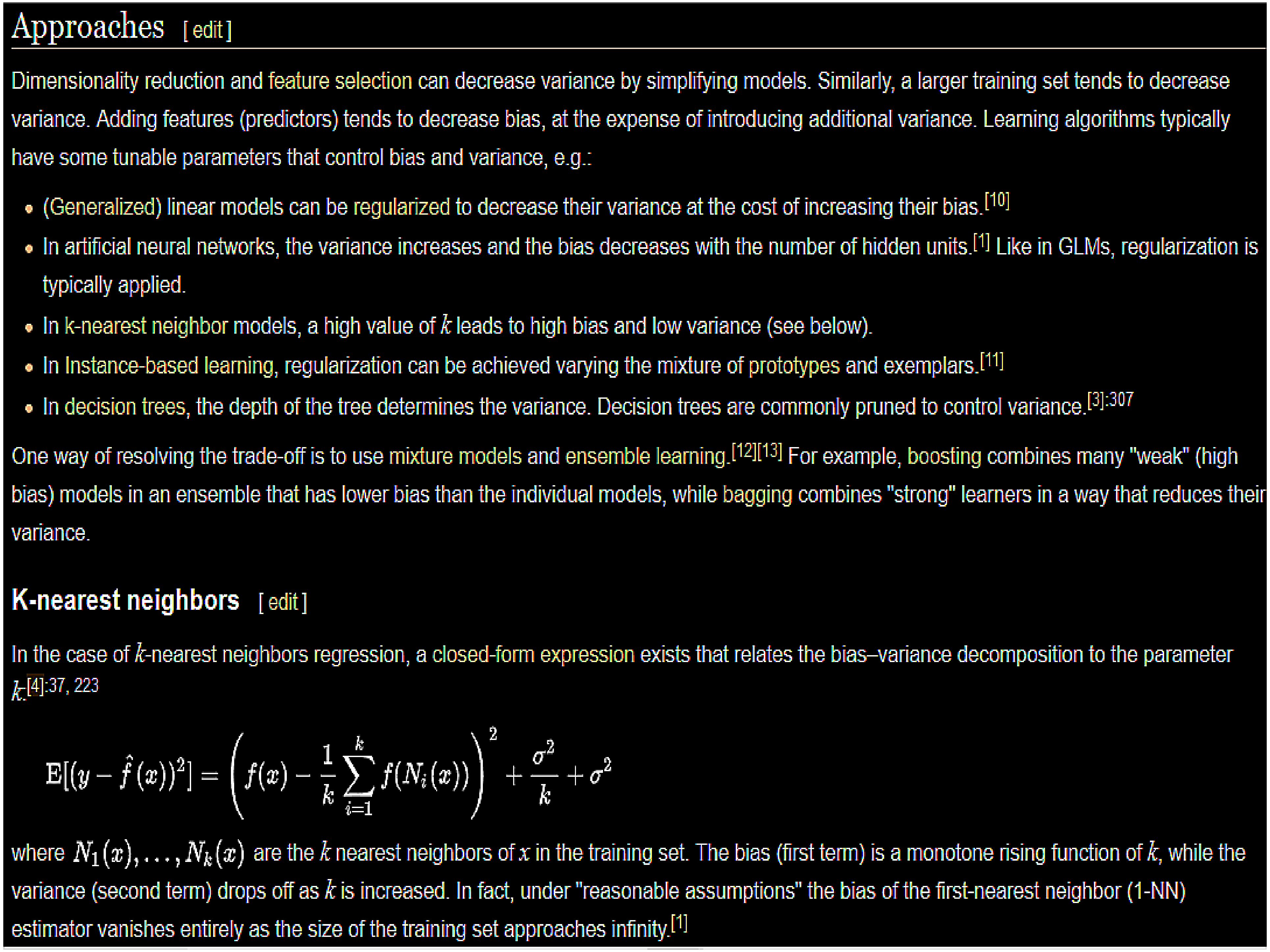
This, given $y = f + \epsilon$ and $\mathrm{E}[\epsilon] = 0$, implies $\mathrm{E}[y] = \mathrm{E}[f + \epsilon] = \mathrm{E}[f] = f$.

Also, since $\mathrm{Var}[\epsilon] = \sigma^2$

$$\mathrm{Var}[y] = \mathrm{E}[(y - \mathrm{E}[y])^2] = \mathrm{E}[(y - f)^2] = \mathrm{E}[(f + \epsilon - f)^2] = \mathrm{E}[\epsilon^2] = \mathrm{Var}[\epsilon] + \mathrm{E}[\epsilon]^2 = \sigma^2$$

Thus, since ϵ and $\hat{f}$ are independent, we can write

$$\begin{aligned}
\mathrm{E}\big[(y - \hat{f})^2\big] &= \mathrm{E}[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= \mathrm{E}[y^2] + \mathrm{E}[\hat{f}^2] - \mathrm{E}[2y\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{E}[y]^2 + \mathrm{Var}[\hat{f}] + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}[\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + (f - \mathrm{E}[\hat{f}])^2 \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + \mathrm{E}[f - \hat{f}]^2 \\
&= \sigma^2 + \mathrm{Var}[\hat{f}] + \mathrm{Bias}[\hat{f}]^2
\end{aligned}$$



Approaches [edit]

Dimensionality reduction and feature selection can decrease variance by simplifying models. Similarly, a larger training set tends to decrease variance. Adding features (predictors) tends to decrease bias, at the expense of introducing additional variance. Learning algorithms typically have some tunable parameters that control bias and variance, e.g.:

- (Generalized) linear models can be regularized to decrease their variance at the cost of increasing their bias.[10]
- In artificial neural networks, the variance increases and the bias decreases with the number of hidden units.[1] Like in GLMs, regularization is typically applied.
- In k-nearest neighbor models, a high value of k leads to high bias and low variance (see below).
- In Instance-based learning, regularization can be achieved varying the mixture of prototypes and exemplars.[11]
- In decision trees, the depth of the tree determines the variance. Decision trees are commonly pruned to control variance.[3]:307

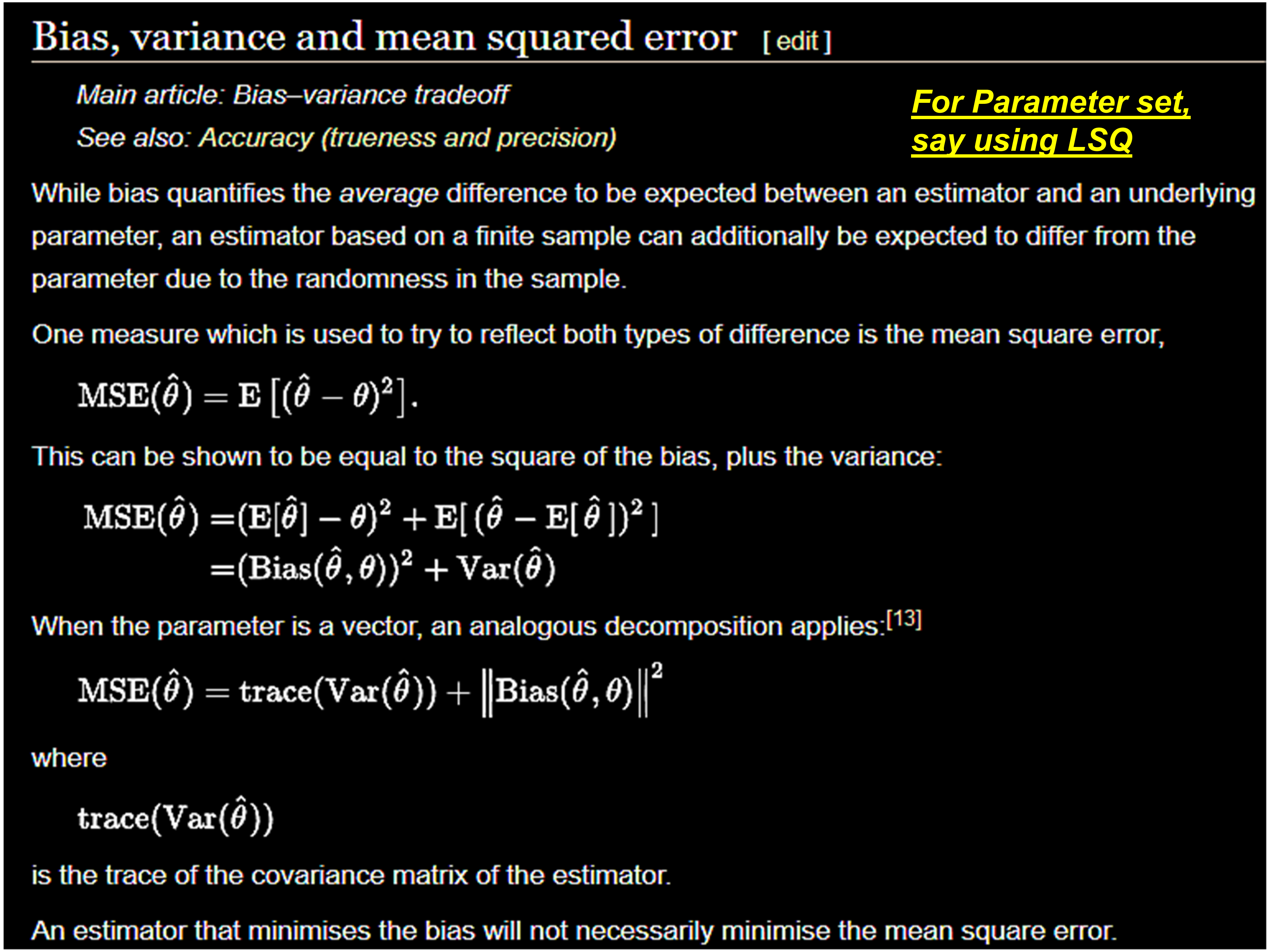
One way of resolving the trade-off is to use mixture models and ensemble learning.[12][13] For example, boosting combines many "weak" (high bias) models in an ensemble that has lower bias than the individual models, while bagging combines "strong" learners in a way that reduces their variance.

K-nearest neighbors [edit]

In the case of k-nearest neighbors regression, a closed-form expression exists that relates the bias–variance decomposition to the parameter k.[4]:37, 223

$$\mathrm{E}[(y-\hat{f}(x))^2] = \left(f(x) - \frac{1}{k}\sum_{i=1}^{k} f(N_i(x))\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

where $N_1(x), \dots, N_k(x)$ are the k nearest neighbors of x in the training set. The bias (first term) is a monotone rising function of k, while the variance (second term) drops off as k is increased. In fact, under "reasonable assumptions" the bias of the first-nearest neighbor (1-NN) estimator vanishes entirely as the size of the training set approaches infinity.[1]



Bias, variance and mean squared error [edit]

Main article: Bias–variance tradeoff

See also: Accuracy (trueness and precision)

For Parameter set, say using LSQ

While bias quantifies the *average* difference to be expected between an estimator and an underlying parameter, an estimator based on a finite sample can additionally be expected to differ from the parameter due to the randomness in the sample.

One measure which is used to try to reflect both types of difference is the mean square error,

$$\operatorname{MSE}(\hat{\theta}) = \operatorname{E}\big[(\hat{\theta}-\theta)^2\big].$$

This can be shown to be equal to the square of the bias, plus the variance:

$$\begin{aligned}\operatorname{MSE}(\hat{\theta}) &= (\operatorname{E}[\hat{\theta}]-\theta)^2 + \operatorname{E}[\,(\hat{\theta}-\operatorname{E}[\,\hat{\theta}\,])^2\,] \\ &= (\operatorname{Bias}(\hat{\theta},\theta))^2 + \operatorname{Var}(\hat{\theta})\end{aligned}$$

When the parameter is a vector, an analogous decomposition applies:[13]

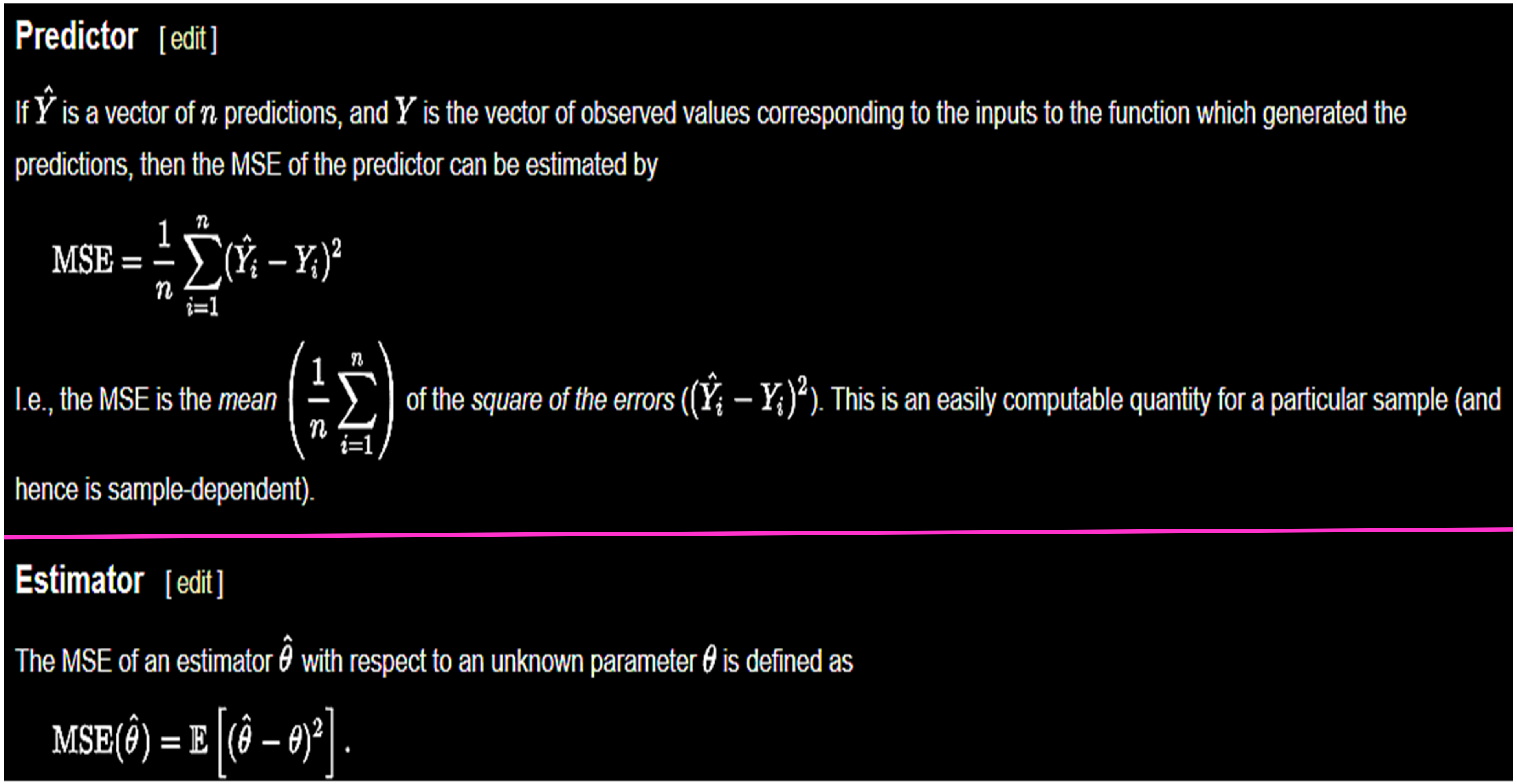
$$\operatorname{MSE}(\hat{\theta}) = \operatorname{trace}(\operatorname{Var}(\hat{\theta})) + \left\|\operatorname{Bias}(\hat{\theta},\theta)\right\|^2$$

where

$$\operatorname{trace}(\operatorname{Var}(\hat{\theta}))$$

is the trace of the covariance matrix of the estimator.

An estimator that minimises the bias will not necessarily minimise the mean square error.



Predictor [edit]

If $\hat{Y}$ is a vector of n predictions, and Y is the vector of observed values corresponding to the inputs to the function which generated the predictions, then the MSE of the predictor can be estimated by

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

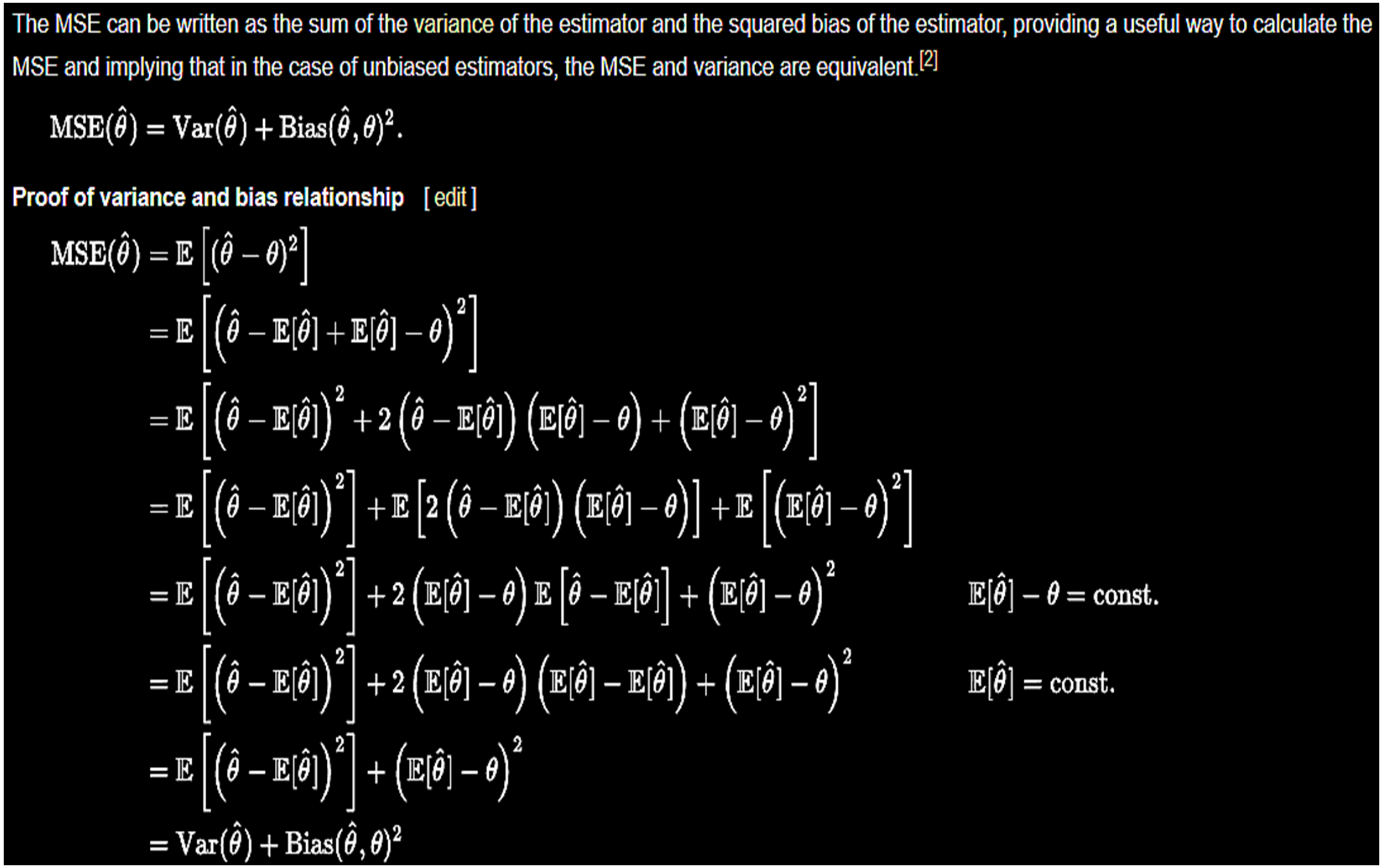
I.e., the MSE is the *mean* $\left(\frac{1}{n}\sum_{i=1}^{n}\right)$ of the *square of the errors* $((\hat{Y}_i - Y_i)^2)$. This is an easily computable quantity for a particular sample (and hence is sample-dependent).

Estimator [edit]

The MSE of an estimator $\hat{\theta}$ with respect to an unknown parameter θ is defined as

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right].$$

SRC - WIKI



The MSE can be written as the sum of the variance of the estimator and the squared bias of the estimator, providing a useful way to calculate the MSE and implying that in the case of unbiased estimators, the MSE and variance are equivalent.[2]

$$\operatorname{MSE}(\hat{\theta}) = \operatorname{Var}(\hat{\theta}) + \operatorname{Bias}(\hat{\theta},\theta)^2.$$

Proof of variance and bias relationship [edit]

$$\begin{aligned}
\operatorname{MSE}(\hat{\theta}) &= \mathbb{E}\left[(\hat{\theta}-\theta)^2\right] \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]+\mathbb{E}[\hat{\theta}]-\theta\right)^2\right] \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)^2 + 2\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)\left(\mathbb{E}[\hat{\theta}]-\theta\right) + \left(\mathbb{E}[\hat{\theta}]-\theta\right)^2\right] \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)^2\right] + \mathbb{E}\left[2\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)\left(\mathbb{E}[\hat{\theta}]-\theta\right)\right] + \mathbb{E}\left[\left(\mathbb{E}[\hat{\theta}]-\theta\right)^2\right] \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)^2\right] + 2\left(\mathbb{E}[\hat{\theta}]-\theta\right)\mathbb{E}\left[\hat{\theta}-\mathbb{E}[\hat{\theta}]\right] + \left(\mathbb{E}[\hat{\theta}]-\theta\right)^2 && \mathbb{E}[\hat{\theta}]-\theta = \text{const.} \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)^2\right] + 2\left(\mathbb{E}[\hat{\theta}]-\theta\right)\left(\mathbb{E}[\hat{\theta}]-\mathbb{E}[\hat{\theta}]\right) + \left(\mathbb{E}[\hat{\theta}]-\theta\right)^2 && \mathbb{E}[\hat{\theta}] = \text{const.} \\
&= \mathbb{E}\left[\left(\hat{\theta}-\mathbb{E}[\hat{\theta}]\right)^2\right] + \left(\mathbb{E}[\hat{\theta}]-\theta\right)^2 \\
&= \operatorname{Var}(\hat{\theta}) + \operatorname{Bias}(\hat{\theta},\theta)^2
\end{aligned}$$

$$\begin{aligned}\text{MSE}(x_0) &= \text{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\ &= \text{E}_{\mathcal{T}}[\hat{y}_0 - \text{E}_{\mathcal{T}}(\hat{y}_0)]^2 + [\text{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\ &= \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0). \qquad (2.25)\end{aligned}$$

$Y = X^T\beta + \varepsilon$ $\varepsilon \sim N(0, \sigma^2)$

$$\begin{aligned}\text{EPE}(x_0) &= \text{E}_{y_0|x_0}\text{E}_{\mathcal{T}}(y_0 - \hat{y}_0)^2 \\ &= \text{Var}(y_0|x_0) + \text{E}_{\mathcal{T}}[\hat{y}_0 - \text{E}_{\mathcal{T}}\hat{y}_0]^2 + [\text{E}_{\mathcal{T}}\hat{y}_0 - x_0^T\beta]^2 \\ &= \text{Var}(y_0|x_0) + \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0) \\ &= \sigma^2 + \text{E}_{\mathcal{T}}x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0\sigma^2 + 0^2. \qquad (2.27)\end{aligned}$$

$$\begin{aligned}\text{E}_{x_0}\text{EPE}(x_0) &\sim \text{E}_{x_0}x_0^T\text{Cov}(X)^{-1}x_0\sigma^2/N + \sigma^2 \\ &= \text{trace}[\text{Cov}(X)^{-1}\text{Cov}(x_0)]\sigma^2/N + \sigma^2 \\ &= \sigma^2(p/N) + \sigma^2. \qquad (2.28)\end{aligned}$$

$\text{E}(X) = 0$, then $\mathbf{X}^T\mathbf{X} \to N\text{Cov}(X)$

<u>Equation 2.25 Derivation;</u> E_τ *is expectation over training set*

$$MSE(x_0) = E_\tau([f(x_0) - \hat{y}_0\,]^2)$$
$$= E_\tau \;\; [f(x_0)^2 - 2f(x_0)\hat{y}_0 \; + \; \hat{y}_0^2]$$
$$= E_\tau \;\; [\hat{y}_0^2] - 2E_T[f(x_0)]E_T[\hat{y}_0] \; + E_T[f(x_0)^2]$$

Add and subtract $(E_T[\hat{y}_0])^2$

$$LHS = \; E_T[\hat{y}_0^2] - (E_T[\hat{y}_0])^2 + \; (E_T[\hat{y}_0])^2 - 2E_T[f(x_0)]E_T[\hat{y}_0] \; + E_T[f(x_0)^2]$$

Since, f is deterministic (given a particular input, will always provide the same output), $E_T[f(x_0)] = f(x_0)$ and

$$Var(A) = E[(A - E[A])^2] = E[A^2] - (E[A])^2$$

$$\text{LHS} \; = E_T\big[\hat{y}_0 - E_T[\hat{y}_0]\big]^2 + \big(E_T[\hat{y}_0] - f(x_0)\big)^2$$
$$= Var_T(\hat{y}_0) + Bias^2(\hat{y}_0)$$

$$MSE(x_0) = Var_T(\hat{y}_0) + Bias^2(\hat{y}_0) \qquad\qquad (\mathbf{2.25})$$

Equation 2.27 and 2.28 Derivation

Suppose the relation between Y and X is linear,

$$Y = X^T\beta + \epsilon$$

where $\epsilon \sim N(0, \sigma^2)$ and we fit a model by least squares to the training data.

For an arbitrary point x_0, we have $\hat{y}_0 = x_0^T\hat{\beta}$, which can be written as $\hat{y}_0 = x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i$, where $l_i(x_0)$ is the i-th element of $X(X^TX)^{-1}x_0$. Since under this model the least squares estimates are unbiased, we find that:

$$EPE(x_0) = E_{y_0|x_0}E_T(y_0 - \hat{y}_0)^2$$

$$= E_{y_0|x_0}E_T[y_0^2 - 2y_0\hat{y}_0 + \hat{y}_0^2]$$

$$= E_{y_0|x_0}[y_0^2 - 2y_0E_T(\hat{y}_0) + E_T(\hat{y}_0^2)]$$

$$= E_{y_0|x_0}[y_0^2] - 2E_T(\hat{y}_0)E_{y_0|x_0}(y_0) + E_T(\hat{y}_0^2)$$

$$EPE(x_0) = E_{y_0|x_0} E_T (y_0 - \hat{y}_0)^2$$
$$= E_{y_0|x_0} E_T [y_0^2 - 2y_0\hat{y}_0 + \hat{y}_0^2]$$
$$= E_{y_0|x_0} [y_0^2 - 2y_0 E_T(\hat{y}_0) + E_T(\hat{y}_0^2)]$$
$$= E_{y_0|x_0}[y_0^2] - 2E_T(\hat{y}_0) E_{y_0|x_0}(y_0) + E_T(\hat{y}_0^2)$$

Using $E(A^2) = Var(A) + [E(A)]^2$ on above, we have, EPE(x_0)

$$= Var(y_0|x_0) + \left(E_{y_0|x_0}[y_0]\right)^2 - 2E_T(\hat{y}_0) E_{y_0|x_0}(y_0) + E_T(\hat{y}_0^2)$$

Add and subtract $(E_T[\hat{y}_0])^2$; and $E_{y_0|x_0}[y_0] = y_0 = x_0^T\beta$

$$EPE(x_0)$$
$$= Var(y_0|x_0) + E_T(\hat{y}_0^2) - (E_T[\hat{y}_0])^2 + (E_T[\hat{y}_0])^2 - 2y_0 E_T(\hat{y}_0) + y_0^2$$
$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - y_0)^2$$

$$\boxed{\begin{aligned} &= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - x_0^T\beta)^2 \\ &= Var(y_0|x_0) + Var_T(\hat{y}_0) + Bias^2(\hat{y}_0) \end{aligned}}$$

$$EPE(x_0)$$
$$= Var(y_0|x_0) + E_T(\hat{y}_0^2) - (E_T[\hat{y}_0])^2 + (E_T[\hat{y}_0])^2 - 2y_0E_T(\hat{y}_0) + y_0^2$$
$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - y_0)^2$$
$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - x_0^T\beta)^2$$
$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + Bias^2(\hat{y}_0)$$

Using $Var(y_0|x_0) = \sigma^2$ and substituting the expansion for $\hat{y}_0$

$$EPE(x_0) = \sigma^2 + Var_T\left(x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i\right) + \left(E_T\left[x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i\right] - x_0^T\beta\right)^2$$
$$= \sigma^2 + Var_T\left(x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i\right) + \left(x_0^T\beta + E_T\left[\sum_{i=1}^{N} l_i(x_0)\epsilon_i\right] - x_0^T\beta\right)$$

Since $E[\epsilon] = 0$, $E_T\left[\sum_{i=1}^{N} l_i(x_0)\epsilon_i\right] = 0$

$$EPE(x_0) = \sigma^2 + Var_T\left(x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i\right) + 0$$

Since $E[\epsilon] = 0$, $E_T\left[\sum_{i=1}^{N} l_i(x_0)\epsilon_i\right] = 0$

$$EPE(x_0) = \sigma^2 + Var_T\left(x_0^T\beta + \sum_{i=1}^{N} l_i(x_0)\epsilon_i\right) + 0$$

$$EPE(x_0) = \sigma^2 + Var_T(x_0^T\beta) + Var_T\left(\sum_{i=1}^{N} l_i(x_0)\epsilon_i\right)$$

Since $Var_T(x_0^T\beta) = 0$,

$$= \sigma^2 + \sum_{i=1}^{N} l_i^2(x_0)\,\sigma^2$$

$$= \sigma^2 + x_0^T(X^TX)^{-T}X^TX(X^TX)^{-1}x_0\,\sigma^2$$

Since X^TX is symmetric

$$\boxed{EPE(x_0) = \sigma^2 + x_0^T(X^TX)^{-1}x_0\,\sigma^2 \qquad \mathbf{(2.27)}}$$

Since $X^T X$ is symmetric

$$EPE(x_0) = \sigma^2 + x_0^T (X^T X)^{-1} x_0 \, \sigma^2 \qquad \mathbf{(2.27)}$$

Here, we have incurred an additional variance σ^2 in the prediction error, since our target is not deterministic. There is no bias, and the variance depends on x_0.

If N is large and T were selected at random, and assuming $E(X) = 0$, then $X^T X \rightarrow N\, Cov\,(X)$ and

$$E_{x_0} EPE(x_0) \sim \frac{E_{x_0}\left(x_0^T \left(Cov(X)\right)^{-1} x_0\right) \sigma^2}{N} + \sigma^2$$

Using property, $\boldsymbol{E}\left(\boldsymbol{U}^{\boldsymbol{T}} \boldsymbol{V} \boldsymbol{U}\right) = E\left(tr(VUU^T)\right) = \boldsymbol{tr}\left(\boldsymbol{V}\left(\boldsymbol{E}\left(\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}}\right)\right)\right)$

$$= \frac{trace[Cov(X)^{-1} Cov(x_0)]\sigma^2}{N} + \sigma^2$$

$$\boxed{E_{x_0} EPE(x_0) \sim \sigma^2 \left(\frac{p}{N}\right) + \sigma^2 \qquad \mathbf{(2.28)}}$$

$$E_{x_0}EPE(x_0) = \frac{trace[Cov(X)^{-1}Cov(x_0)]\sigma^2}{N} + \sigma^2$$

$$E_{x_0}EPE(x_0) \sim \sigma^2\left(\frac{p}{N}\right) + \sigma^2 \qquad \textbf{(2.28)}$$

Here we see that the expected EPE increases linearly as a function of p, with slope σ^2/N. If N is large and/or σ^2 is small, this growth in variance is negligible (0 in the deterministic case).

By imposing some heavy restrictions on the class of models being fitted, we have avoided the curse of dimensionality.

model $Y = f(X) + \varepsilon$, with $\mathrm{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$.

$$\begin{aligned}
\mathrm{EPE}_k(x_0) &= \mathrm{E}[(Y - \hat{f}_k(x_0))^2 | X = x_0] \\
&= \sigma^2 + [\mathrm{Bias}^2(\hat{f}_k(x_0)) + \mathrm{Var}_{\mathcal{T}}(\hat{f}_k(x_0))] && (2.46) \\
&= \sigma^2 + \left[f(x_0) - \frac{1}{k} \sum_{\ell=1}^{k} f(x_{(\ell)}) \right]^2 + \frac{\sigma^2}{k}. && (2.47)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + [\mathrm{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\
&= \text{Irreducible Error} + \mathrm{Bias}^2 + \text{Variance}. && (7.9)
\end{aligned}$$

Equation 2.47 Derivation

The k-nearest-neighbor regression fit $\hat{f}_k(x_0)$ usually illustrates the competing forces that effect the predictive ability of such approximation. Suppose the data arise from a model $Y = f(X) + \epsilon$, with $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma^2$.

For simplicity here we assume that the values of x_i in the samples are fixed in advance (nonrandom). For an input x_0,

$$\hat{y}_0 = \hat{f}_k(x_0) = \frac{1}{k}\sum_{l=1}^{k} y_{(l)} = \frac{1}{k}\sum_{l=1}^{k}\left[f\left(x_{(l)}\right) + \epsilon_{(l)}\right]$$

The subscript in the parentheses-(l) indicate the sequence of nearest neighbors to x_0. The expected prediction error at x_0, also known as *test* or *generation* error, can be decomposed:

$$EPE_k(x_0) = E\left[\left(Y - \hat{f}_k(x_0)\right)^2 \middle| X = x_0\right]$$

$$EPE_k(x_0) = E\left[\left(Y - \hat{f}_k(x_0)\right)^2 \middle| X = x_0\right]$$

Using $EPE(x_0)$

$= Var(y_0|x_0) + E_T(\hat{y}_0^2) - (E_T[\hat{y}_0])^2 + (E_T[\hat{y}_0])^2 - 2y_0 E_T(\hat{y}_0) + y_0^2$

$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - y_0)^2$$

$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + (E_T[\hat{y}_0] - x_0^T\beta)^2$$

$$= Var(y_0|x_0) + Var_T(\hat{y}_0) + Bias^2(\hat{y}_0)$$

$$EPE(x_0) = \sigma^2 + \left[Bias^2\left(\hat{f}_k(x_0)\right) + Var_T\left(\hat{f}_k(x_0)\right)\right] \qquad \mathbf{(2.46)}$$

$$= \sigma^2 + \left(y_0 - E_T[\hat{f}_k(x_0)]\right)^2 + Var_T[\hat{f}_k(x_0)]$$

$$= \sigma^2 + \left(y_0 - E_T\left[\frac{1}{k}\sum_{l=1}^{k}[f\left(x_{(l)}\right) + \epsilon_{(l)}]\right]\right)^2 + Var_T\left[\frac{1}{k}\sum_{l=1}^{k}[f\left(x_{(l)}\right) + \epsilon_{(l)}]\right]$$

$$EPE(x_0) = \sigma^2 + \left[Bias^2\left(\hat{f}_k(x_0)\right) + Var_T\left(\hat{f}_k(x_0)\right)\right] \qquad \mathbf{(2.46)}$$

$$= \sigma^2 + \left(y_0 - E_T[\hat{f}_k(x_0)]\right)^2 + Var_T[\hat{f}_k(x_0)]$$

$$= \sigma^2 + \left(y_0 - E_T\left[\frac{1}{k}\sum_{l=1}^{k}\left[f\left(x_{(l)}\right) + \epsilon_{(l)}\right]\right]\right)^2 + Var_T\left[\frac{1}{k}\sum_{l=1}^{k}\left[f\left(x_{(l)}\right) + \epsilon_{(l)}\right]\right]$$

$$= \sigma^2 + \left(f(x_0) - \frac{1}{k}\left(E_T\left[\sum_{l=1}^{k} f\left(x_{(l)}\right)\right] + \cancel{E_T\left[\sum_{l=1}^{k}\epsilon_{(l)}\right]}\right)\right)^2$$

$$+ \frac{1}{k^2}\left(\cancel{Var_T\left[\sum_{l=1}^{k} f\left(x_{(l)}\right)\right]} + Var_T\left(\sum_{l=1}^{k}\epsilon_{(l)}\right)\right)$$

Since,

$\sum_{l=1}^{k} f\left(x_{(l)}\right)$ is a constant:

$$EPE_k(x_0) = \sigma^2 + \left[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f\left(x_{(l)}\right)\right]^2 + \frac{k\sigma^2}{k^2}$$

$$\boldsymbol{EPE_k(x_0) = \sigma^2 + \left[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f\left(x_{(l)}\right)\right]^2 + \frac{\sigma^2}{k}} \qquad \mathbf{(2.47)}$$

7.3 The Bias–Variance Decomposition

As in Chapter 2, if we assume that $Y = f(X) + \varepsilon$ where $\mathrm{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma_\varepsilon^2$, we can derive an expression for the expected prediction error of a regression fit $\hat{f}(X)$ at an input point $X = x_0$, using squared-error loss:

$$\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + [\mathrm{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)]^2 \\
&= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\
&= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}. \qquad (7.9)
\end{aligned}$$

The first term is the variance of the target around its true mean $f(x_0)$, and cannot be avoided no matter how well we estimate $f(x_0)$, unless $\sigma_\varepsilon^2 = 0$. The second term is the squared bias, the amount by which the average of our estimate differs from the true mean; the last term is the variance; the expected squared deviation of $\hat{f}(x_0)$ around its mean. Typically the more complex we make the model $\hat{f}$, the lower the (squared) bias but the higher the variance.

For the k-nearest-neighbor regression fit, these expressions have the simple form

$$\begin{aligned}
\mathrm{Err}(x_0) &= E[(Y - \hat{f}_k(x_0))^2 | X = x_0] \\
&= \sigma_\varepsilon^2 + \left[f(x_0) - \frac{1}{k}\sum_{\ell=1}^{k} f(x_{(\ell)}) \right]^2 + \frac{\sigma_\varepsilon^2}{k}. \qquad (7.10)
\end{aligned}$$

BYE