

Foundation of Machine Learning (CSE4032)

Lecture 02: Overview of Supervised Learning (Part 1)

Dr. Kundan Kumar
Associate Professor
Department of ECE



Faculty of Engineering (ITER)
S'O'A Deemed to be University, Bhubaneswar, India-751030
© 2020 Kundan Kumar, All Rights Reserved

Outline

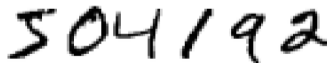
- 1 Introduction
- 2 Statistical Learning
- 3 Estimate f
- 4 Least Square Method
- 5 Nearest-Neighbor Method
- 6 Model accuracy
- 7 References

Introduction

- Goal is to use input to predict outputs.
 - Inputs are measured or preset
 - Inputs have some influence on one or more outputs
- In statistics, the term inputs and predictors will be used interchangeably.
- We call it **independent variable** in more classy way.
- In pattern classification we call it **features**.
- Outputs are called **responses** or the **dependent variable**.

Variable Types

- Output/Target can be
 - **quantitative** (numerical values)
 - **qualitative** (categorical values, factor, discrete variables)
 - Two class: $\{0, 1\}$ or $\{-1, 1\}$
 - Multiclass: dummy variable, K-level qualitative variable represented by k-bits.
- Examples of categorical variable
 - Iris discrimination
 - Species of Iris, $\mathcal{G} = \{Virginia, Setosa, \text{ and } Versicolor\}$
 - Handwritten digit recognition
 - Output is one of 10 different digit classes: $\mathcal{G} = \{0, 1, \dots, 9\}$

A handwritten image showing the digits '504192' in black ink on a white background. The digits are slightly slanted and have a casual, cursive-like appearance.

- There is **no explicit ordering** in the classes.

Variable Types

- Based on these, naming convention for prediction tasks
 - **Regression**, when we predict quantitative outputs.
 - **Classification**, when we predict qualitative outputs.
- Input may also vary in measurement type: qualitative and quantitative variable.
- Third variable: **ordered categorical**. Ex - small, medium, large

Standard Notations

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

- n : number of distinct data points/observations
- p : number of variables available for making predictions
- x_{ij} : j th variable for the i th observation, where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, p$
- i : index for samples / observations
- j : index for variables
- \mathbf{X} : denote a $n \times p$ matrix whose (i, j) th element is x_{ij}

Standard Notations

x_i : i th row of \mathbf{X} having length p , $[x_{i1}, x_{i2}, \dots, x_{ip}]^T$, i th observation

\mathbf{x}_j : j th column of \mathbf{X} having length n , $[x_{1j}, x_{2j}, \dots, x_{nj}]^T$, j th variable

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}$$

The T notation denotes the transpose of a matrix or vector.

$$\mathbf{X}^T = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & & \vdots \\ x_{1p} & x_{2p} & \cdots & x_{np} \end{pmatrix} \qquad x_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}$$

Standard Notations

y_i : i th observation of the variable

\mathbf{y} : set of n observation in a vector form

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- Then $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ is a set of observed data.
- A vector of length n will always be denoted in lower case bold, e.g. \mathbf{x} .
- A vector of length p will be denoted in lower case normal/italic font, e.g. x
- Scalars will also be denoted in lower case normal font, e.g. a .

Standard Notations

- **Matrices** will be denoted using bold capitals, such as \mathbf{A} .
- **Random variables** will be denoted using capital normal font, e.g. A , regardless of their dimensions.
- To indicate that an object is a **scalar**, we will use the notation $a \in \mathbb{R}$.
- To indicate that it is a **vector** of length k , we will use $a \in \mathbb{R}^k$ (or $a \in \mathbb{R}^n$ if it is of length n).
- We will indicate that an object is a $r \times s$ matrix using $\mathbf{A} \in \mathbb{R}^{r \times s}$
- Suppose that $\mathbf{A} \in \mathbb{R}^{r \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times s}$. Then the product of \mathbf{A} and \mathbf{B} is denoted as \mathbf{AB} and (i, j) th element is

$$(\mathbf{AB})_{ij} = \sum_{k=1}^d a_{ik} b_{kj}$$

Statistical learning problem

- Suppose that we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product.
- It is not possible for our client to directly increase sales of the product. They can control the advertising expenditure in each of the three media (TV, Radio, and Newspaper).
- Therefore, if we determine that there is an association between advertising and sales, then we can instruct our client to adjust advertising budgets, thereby indirectly increasing sales.
- Our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets.

Dataset

- The **Advertising data set**:
 - consists of the sales of that product in 200 different markets
 - along with advertising budgets for the product in each of those markets for three different media: **TV**, **radio**, and **newspaper**.

X1	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

- In this example, the advertising budgets are **input variables** while sales is an **output variable**.

Variables

- Input variable
 - typically denoted using the symbol X , with a subscript to distinguish them.
 - X_1 might be the TV budget, X_2 the radio budget, and X_3 the newspaper budget.
 - The inputs go by different names, such as predictors, independent variables, features, or sometimes just variables.
- Output variable
 - in this case, sales
 - often called the response or dependent variable, and is typically denoted using the symbol Y .

Statistical Model

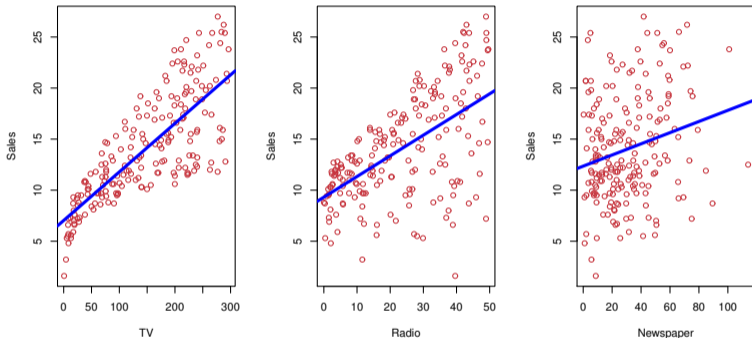


Figure: Advertising data

- Perhaps, we can do better using a model

$$sales \approx f(TV, Radio, Newspaper)$$

- Shown are Sales vs. TV, Radio, and Newspaper, with a blue linear-regression line fit separately to each.
- Can we predict Sales using these three?

Statistical Model

- More generally, suppose that we observe a quantitative response Y and p different predictors, X_1, X_2, \dots, X_p .
- We assume that there is some relationship between Y and $X = (X_1, X_2, \dots, X_p)$, which can be written in the very general form

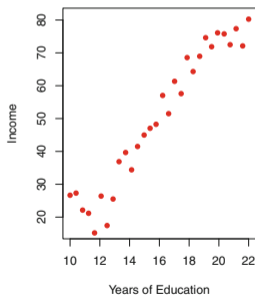
$$Y = f(X) + \epsilon$$

where ϵ captures **measurement errors and other discrepancies**, which is independent of X and has **mean zero**.

- In this formulation, f represents the **systematic** information that X provides about Y .

Relationship between variables

- **Income data set:** Single input variable
 - Left-hand Figure, a plot of income versus years of education for 30 individuals.
 - The plot suggests that one might be able to predict income using years of education.



X1	Education	Income
1	10.00000	26.65884
2	10.40134	27.30644
3	10.84281	22.13241
4	11.24415	21.16984
5	11.64548	15.19263
6	12.08696	26.39895
7	12.48829	17.43531

Figure: Income data set

Relationship between variables

- **Income data set:** Single input variable
 - Left-hand Figure, a plot of income versus years of education for 30 individuals.
 - The plot suggests that one might be able to predict income using years of education.

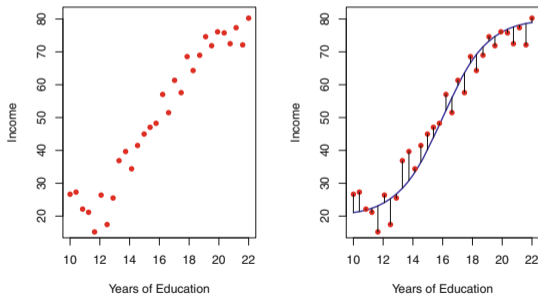


Figure: Income data set

- However, the function f that connects the input variable to the output variable is **in general unknown**.
- One must estimate f based on the observed points.
- The vertical lines represent the error terms ϵ .

Model Estimation, f

- In general, the function f may involve more than one input variable.
- We plot income as a function of **education** and **seniority**.
- Here f is a two-dimensional surface that must be estimated based on the observed data.
- In essence, statistical learning refers to a set of approaches for estimating f .

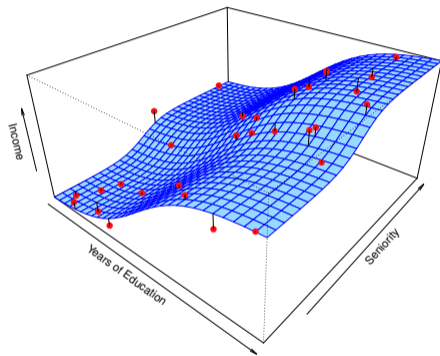


Figure: Income data set

$f(X)$ is good for?

- With a good f , we can make predictions of Y at new points $X = x$
- We can understand which components of $X = (X_1, X_2, \dots, X_p)$ are important in explaining Y , and which are irrelevant. e.g. Seniority and Years of Education have a big impact on Income, but Marital Status typically does not.
- Depending on the complexity of f , we may be able to understand how each component X_j of X affects Y .

Why Estimate f ?

- There are two main reasons that we may wish to estimate f :
 - prediction
 - inference
- **Prediction:** In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained. In this setting, since the error term averages to zero, we can predict Y using

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

where \hat{f} represents our estimate for f , and \hat{Y} represents the resulting prediction for Y .

- In this setting, \hat{f} is often treated as a black box, in the sense that one is not typically concerned with the exact form of \hat{f} , provided that it yields accurate predictions for Y .

Why Estimate f ?

- As an example, suppose that X_1, \dots, X_p are characteristics of a patient's blood sample that can be easily measured in a lab, and Y is a variable encoding the patient's risk for a severe adverse reaction to a particular drug.
- It is natural to seek to predict Y using X to avoid giving the drug in question to patients.
- The accuracy of \hat{Y} as a prediction for Y depends on two quantities, which we will call the **reducible error** and the **irreducible error**.
- In general, \hat{f} will **not be a perfect estimate** for f , and this inaccuracy will introduce some error.
- This error is **reducible** because we can potentially improve the accuracy of \hat{f} by using the most appropriate statistical learning technique to estimate f .
- Y is also a function of ϵ so no matter how well we estimate f , we cannot reduce the error introduced by ϵ known as **irreducible** error.

Why Estimate f ?

- Consider a given estimate \hat{f} and a set of predictors X , which yields the prediction $\hat{Y} = \hat{f}(X)$. Then

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}}, \end{aligned}$$

where $E(Y - \hat{Y})^2$ represents the average, or **expected value**, of the squared difference between the predicted and actual value of Y , and $\text{Var}(\epsilon)$ represents the **variance associated with the error** term ϵ .

- The aim of this course is on techniques for estimating f with the aim of minimizing the reducible error.

Why Estimate f ?

- We are often interested in understanding the way that Y is affected as X_1, \dots, X_p change.
- In this situation we wish to estimate f , but our goal is not necessarily to make predictions for Y .
- We instead want to understand the relationship between X and Y , or more specifically, to understand how Y changes as a function of X_1, \dots, X_p .
- Now, \hat{f} cannot be treated as a black box, because we need to know its exact form. In this setting, one may be interested in answering the following questions:
 - Which predictors are **associated** with the response?
 - What is the relationship between the response and each predictor?
 - Can the **relationship** between Y and each predictor be adequately **summarized** using a linear equation, or is the relationship more complicated?

Why Estimate f ?

- In contrast, consider the Advertising data. One may be interested in answering questions such as:
 - Which media contribute to sales?
 - Which media generate the biggest boost in sales? or
 - How much increase in sales is associated with a given increase in TV advertising?

X1	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

- Finally, some modeling could be conducted **both for prediction and inference**.

Simple approaches for prediction

- Two Simple Approaches to Prediction:
 - **Least Squares**: the linear model makes **huge assumptions** about structure and yields **stable** but **possibly inaccurate** predictions.
 - **Nearest Neighbors**: the method of k-nearest neighbors makes very **mild structural assumptions**. its predictions are **often accurate** but can be **unstable**.

Linear Models and Least Squares

- Given a vector of inputs $X^T = (X_1, X_2, \dots, X_p)$, we predict the output Y via the model

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

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

$$\hat{Y} = X^T \hat{\beta}$$

- Viewed as a function over the p -dimensional input space, $f(X) = X^T \beta$ is linear, and the gradient $f'(X) = \beta$ is a vector in input space that points in the steepest uphill direction.

How do fit the linear model?

- How do we fit the linear model to a set of training data? There are many different methods, but by far the most popular is the method of **least squares**.
- In this approach, we pick the coefficients β to minimize the **residual sum of squares**

$$\text{RSS}(\beta) = \sum_{i=1}^N (y_i - x_i^T \beta)^2$$

$\text{RSS}(\beta)$ is a **quadratic function of the parameters**, and hence its **minimum always exists**, but may not be unique.

How do fit the linear model?

- The solution is easiest to characterize in matrix notation. We can write

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

where \mathbf{X} is an $N \times p$ matrix with each row an input vector, and \mathbf{y} is an N -vector of the outputs in the training set.

- Differentiating w.r.t. β we get the normal equations

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

- If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution is given by

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

How do fit the linear model?

- The fitted value at the i th input x_i is

$$\hat{y}_i = \hat{y}(x_i) = x_i^T \hat{\beta}.$$

- At an arbitrary input x_0 the prediction is

$$\hat{y}(x_0) = x_0^T \hat{\beta}.$$

- The entire fitted surface is characterized by the $(p + 1)$ parameters $\hat{\beta}$. Intuitively, it seems that we do not need a very large data set to fit such a model.

Example

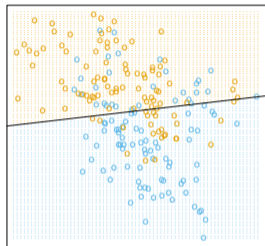


Figure: A classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then fit by linear regression. The line is the decision boundary defined by $x^T \hat{\beta} = 0.5$. The orange shaded region denotes that part of input space classified as ORANGE, while the blue region is classified as BLUE.

- This is an example of the linear model in a classification context.
- The fitted values \hat{Y} are converted to a fitted class variable \hat{G} according to the rule

$$\hat{G} = \begin{cases} \text{ORANGE} & \text{if } \hat{Y} > 0.5 \\ \text{BLUE} & \text{if } \hat{Y} \leq 0.5 \end{cases}$$

- The set of points in \mathbb{R}^2 classified as ORANGE corresponds to $\{x : x^T \hat{\beta} > 0.5\}$, and the two predicted classes are separated by the decision boundary $\{x : x^T \hat{\beta} = 0.5\}$, which is linear in this case.

Example

- Perhaps our linear model is too rigid — or are such errors unavoidable? Remember that these are errors on the training data itself, and we have not said where the constructed data came from. Consider the two possible scenarios:
 - **Scenario 1:** The training data in each class were generated from bivariate Gaussian distributions with uncorrelated components and different means.
 - **Scenario 2:** The training data in each class came from a mixture of 10 low variance Gaussian distributions, with individual means themselves distributed as Gaussian.
- A linear decision boundary is unlikely to be optimal, and in fact is not. The optimal decision boundary is nonlinear and disjoint, and as such will be much more difficult to obtain.

Nearest-Neighbor Methods

- Nearest-neighbor methods use those observations in the training set \mathcal{T} closest in input space to x to form \hat{Y} . Specifically, the k -nearest neighbor fit for \hat{Y} is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training sample.

- Closeness implies a metric, which for the moment we assume is **Euclidean distance**.
- So, in words, we find the k observations with x_i closest to x in input space, and average their responses.

Example

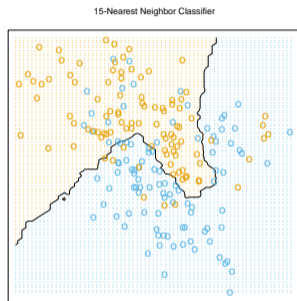


Figure: The same classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1) and then fit by 15-nearest-neighbor averaging. The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.

- We use the same training data as in previous example.
- Use 15-nearest-neighbor averaging of the binary coded response as the method of fitting.
- We see that the decision boundaries that separate the BLUE from the ORANGE regions are far more irregular, and respond to local clusters where one class dominates.

Example

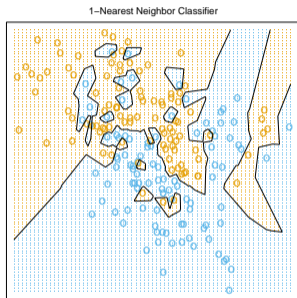


Figure: The same classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.

- \hat{Y} is assigned the value y_ℓ of the closest point x_ℓ to x in the training data.
- In this case the regions of classification can be computed relatively easily, and correspond to a **Voronoi tessellation** of the training data.
- Each point x_i has an associated tile bounding the region for which it is the closest input point.
- For all points x in the tile, $\hat{G}(x) = g_i$. The decision boundary is even more irregular than before.

A comparison between Least Square and k -NN

- It appears that k -nearest-neighbor fits have a single parameter, the number of neighbors k , compared to the p parameters in least-squares fits.
- Although this is the case, we will see that the effective number of parameters of k -nearest neighbors is N/k and is generally bigger than p , and decreases with increasing k .
- To get an idea of why, note that if the neighborhoods were non-overlapping, there would be N/k neighborhoods and we would fit one parameter (a mean) in each neighborhood.
- It is also clear that we cannot use sum-of-squared errors on the training set as a criterion for picking k .

A comparison between Least Square and k -NN

- 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.
- 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 subregion 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.
- Each method has its own situations for which it works best; in particular linear regression is more appropriate for Scenario 1 above, while nearest neighbors are more suitable for Scenario 2.

A comparison between Least Square and k -NN

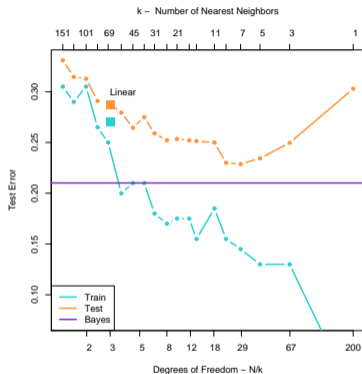


Figure: Misclassification curves for the simulation example used in earlier Figures. A single training sample of size 200 was used, and a test sample of size 10,000. The orange curves are test and the blue are training error for k -nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.

Assessing model accuracy

- There is **no free lunch in statistics**: no one method dominates all others over all possible data sets.
- Hence it is an important task to decide for any given set of data which method produces the best results.
- Evaluating models is one of the best solutions
 - **Regression problem**: measuring the quality of fit

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

where $\hat{f}(x_i)$ is the prediction that \hat{f} gives for the i th observation.

- **Classification problem**

$$\text{Ave} \left(y_0 - \hat{f}(x_0) \right)^2$$

the average squared prediction error for the test observations (x_0, y_0) .

Assessing model accuracy

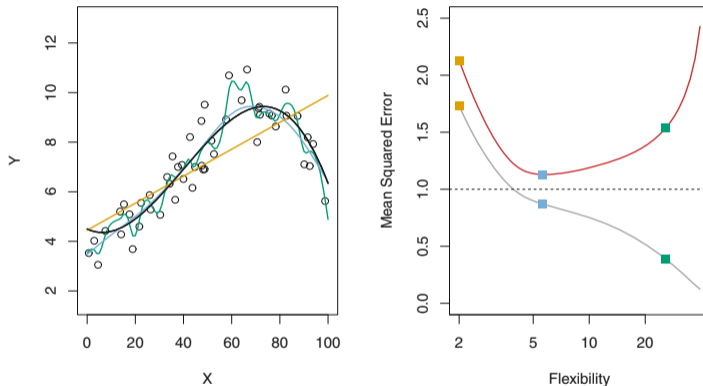


Figure: Left: Data simulated from f , shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Assessing model accuracy

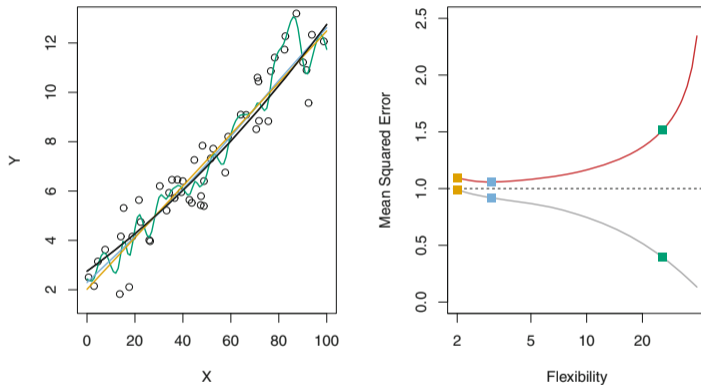


Figure: Left: Data simulated from f , shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Assessing model accuracy

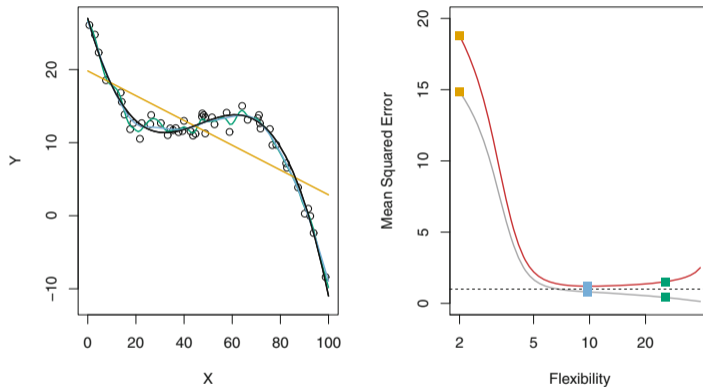


Figure: Left: Data simulated from f , shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

The Bias-Variance Trade-Off

- The U-shape observed in the test MSE curves turns out to be the result of two competing properties of statistical learning methods.
- Suppose the data arise from a model $Y = f(X) + \varepsilon$, with $E(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma^2$.
- For simplicity, here, we assume that the values of x_i in the sample are fixed in advance (nonrandom). The **expected prediction error** at x_0 , also known as **test** or **generalization error**, can be decomposed:

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

The Bias-Variance Trade-Off

- There are three terms in this expression.
 - The first term σ^2 is the irreducible error – the variance of the new test target—and is beyond our control, even if we know the true $f(x_0)$.
 - The second and third terms are under our control, and make up the mean squared error of $\hat{f}_k(x_0)$ in estimating $f(x_0)$, which is broken down into a **bias component** and a **variance component**.
- The bias term is the squared difference between the true mean $f(x_0)$ and the expected value of the estimate- $\left[f(x_0) - E_{\mathcal{T}} \left(\hat{f}_k(x_0) \right) \right]^2$ -where the expectation averages the randomness in the training data.
- This term will most likely increase with k , if the true function is reasonably smooth.
- For small k the few closest neighbors will have values $f(x_{(\ell)})$ close to $f(x_0)$, so their average should be close to $f(x_0)$.

The Bias-Variance Trade-Off

- The variance term is simply the variance of an average here, and decreases as the inverse of k . So as k varies, there is a bias–variance tradeoff.
- What do we mean by the variance and bias of a statistical learning method?
 - Variance refers to the amount by which \hat{f} would change if we estimated it using a different training data set.
 - Since the training data are used to fit the statistical learning method, different training data sets will result in a different \hat{f} .
 - But ideally the estimate for f should not vary too much between training sets.
 - However, if a method has high variance then small changes in the training data can result in large changes in \hat{f} .
 - In general, more flexible statistical methods have higher variance.
- Bias refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model.

The Bias-Variance Trade-Off

- Good test set performance of a statistical learning method requires low variance as well as low squared bias.
- This is referred to as a trade-off because it is easy to obtain a method with extremely low bias but high variance (for instance, by drawing a curve that passes through every single training observation) or a method with very low variance but high bias (by fitting a horizontal line to the data).
- In a real-life situation in which f is unobserved, it is generally not possible to explicitly compute the test MSE, bias, or variance for a statistical learning method.
- Nevertheless, one should always keep the bias-variance trade-off in mind.

The Bias-Variance Trade-Off

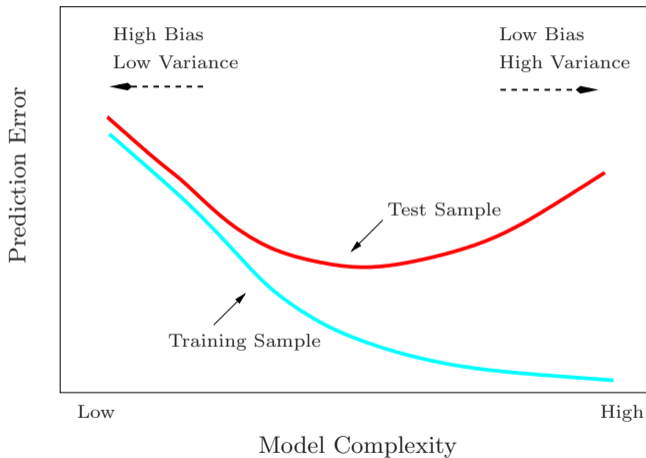




Figure: Test and training error as a function of model complexity.

The Bias-Variance Trade-Off

- More generally, as the model complexity of our procedure is increased, the variance tends to increase and the squared bias tends to decrease. The opposite behavior occurs as the model complexity is decreased. For k -nearest neighbors, the model complexity is controlled by k .
- Typically we would like to choose our model complexity to trade bias off with variance in such a way as to minimize the test error.
- The training error tends to decrease whenever we increase the model complexity, that is, whenever we fit the data harder. However with too much fitting, the model adapts itself too closely to the training data, and will not generalize well (i.e., have large test error). In that case the predictions $\hat{f}(x_0)$ will have large variance.
- In contrast, if the model is not complex enough, it will underfit and may have large bias, again resulting in poor generalization.

References

-  The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Hastie, Tibshirani, and Friedman, Springer.
-  In Introduction to Statistical Learning with Application in R, Second Edition, James, Witten, Hastie, and Tibshirani, Springer.



Thank you!