Machine Learning and Prediction

Zhongjian Lin
University of Georgia

August 21, 2024

Machine Learning

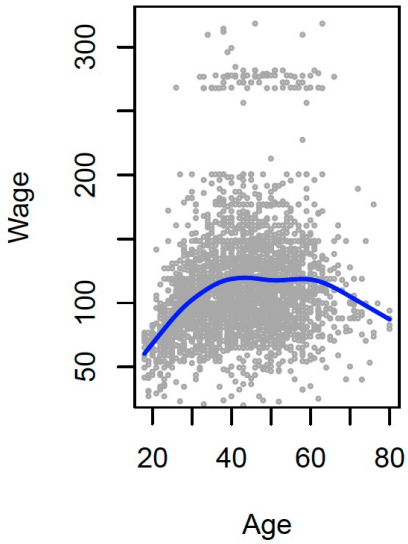
Machine learning refers to a vast set of tools for understanding data. These tools can be classified as supervised or unsupervised.

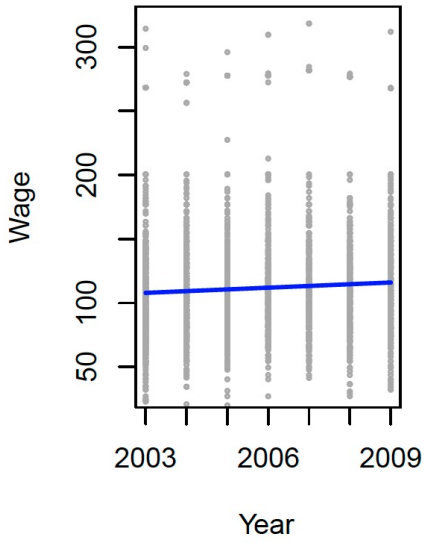
- Output: Wage, Market Direction
- Input: Age, Year, Education Level, Past 5 Days' Percentage changes in the index, etc.

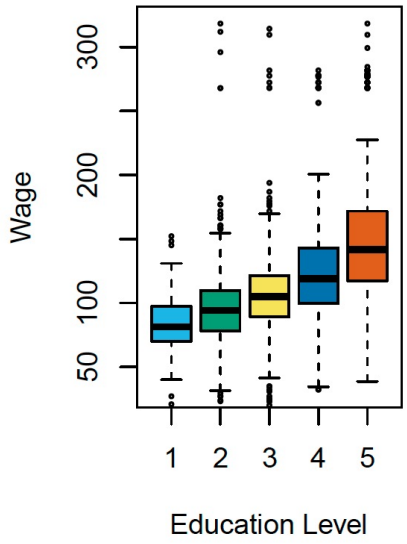
Supervised and Unsupervised Learning

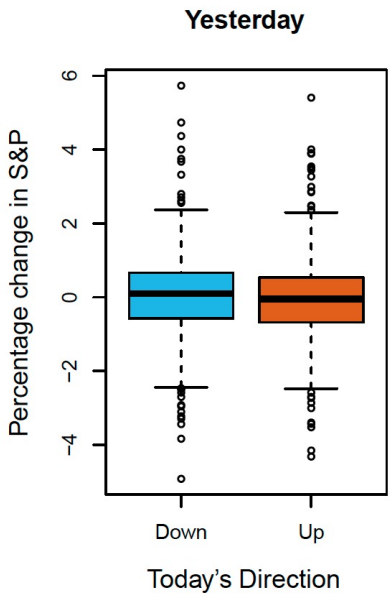
- Supervised Learning: involves building a statistical model for predicting, or estimating, an output based on one or more inputs.
- Unsupervised Learning: there are inputs but no supervising output; nevertheless we can learn relationships and structure from such data.

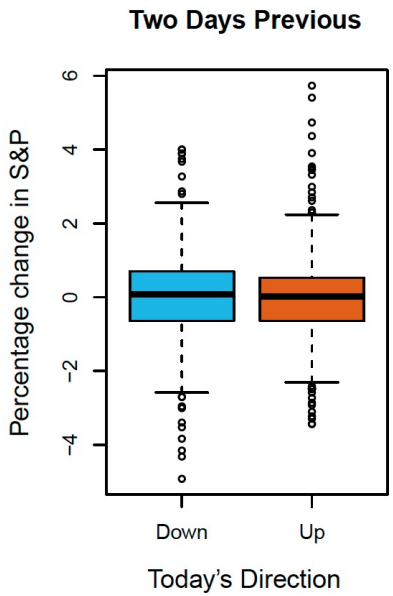
Supervised Learning

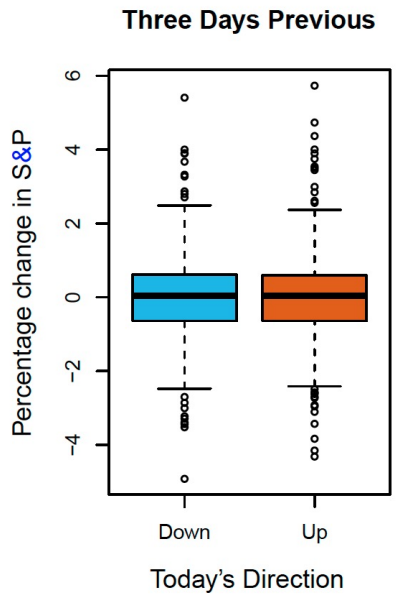




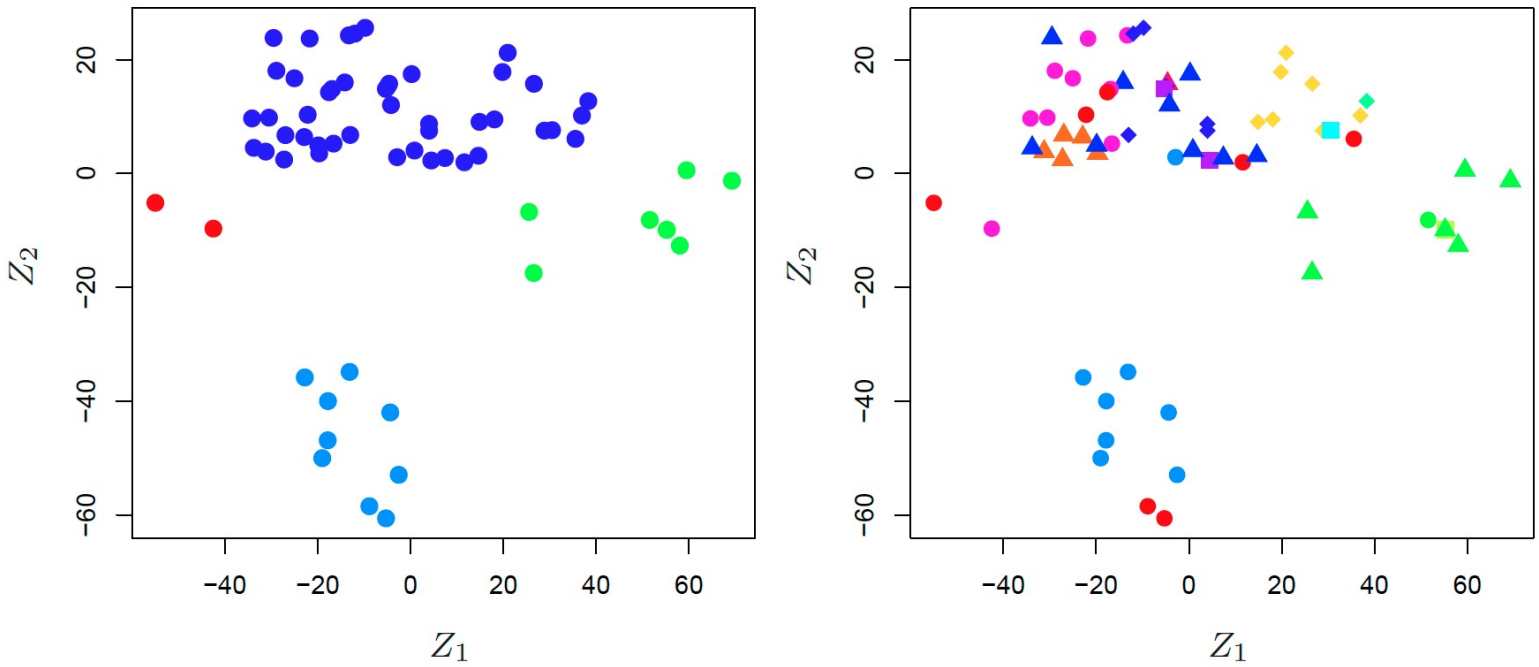








Unsupervised Learning



History of Machine Learning

- At the beginning of the 19th century, the method of least squares was developed, implementing the earliest form of what is now known as linear regression.
- Linear discriminant analysis was proposed in 1936. In the 1940s, various authors put forth an alternative approach, logistic regression.
- In the early 1970s, the term generalized linear model was developed to describe an entire class of machine learning methods that include both linear and logistic regression as special cases.
- By the end of the 1970s, many more techniques for learning from data were available. However, they were almost exclusively linear methods because fitting non-linear relationships was computationally difficult at the time.
- By the 1980s, computing technology had finally improved sufficiently that non-linear methods were no longer computationally prohibitive.
- In the mid 1980s, classification and regression trees were developed, followed shortly by generalized additive models.
- Neural networks gained popularity in the 1980s, and support vector machines arose in the 1990s.
- Natural Language Processing...

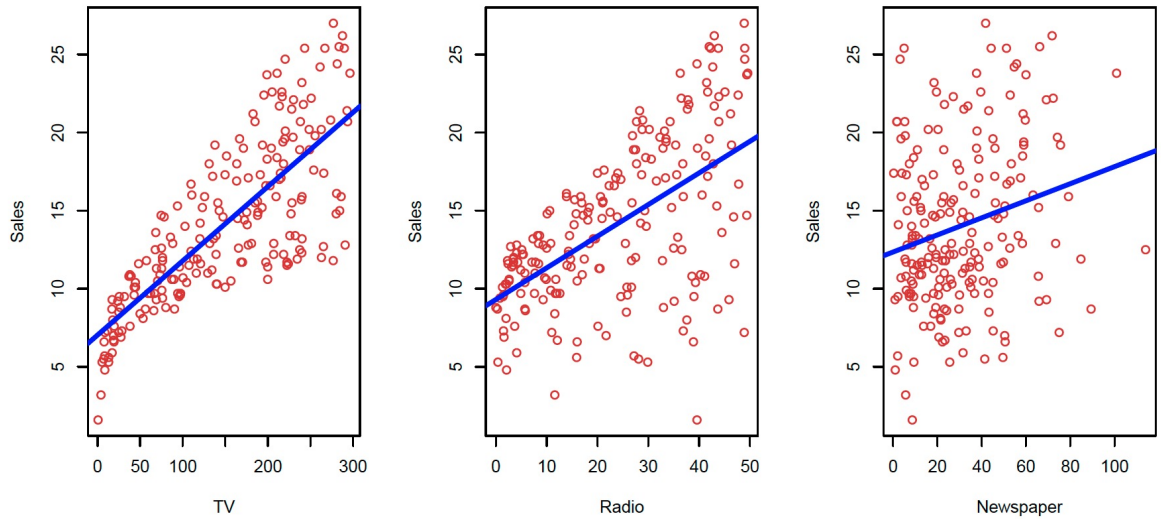
Model

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

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

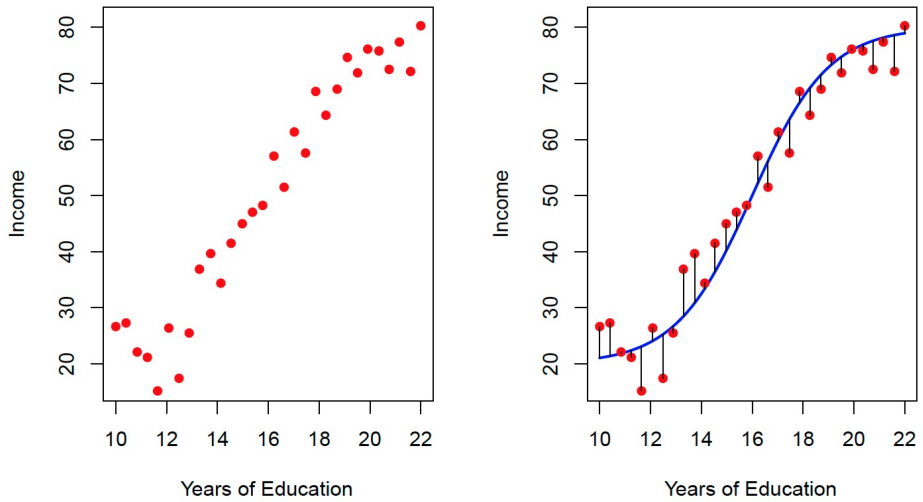
Example: Advertising

Suppose that we are statistical consultants hired by a client to investigate the association between advertising and sales of a particular product. We have *sales* and advertising budgets three different media: *TV, radio, and newspaper* of that product in 200 different markets.



f is some fixed but unknown function of X, and ϵ is a random error term, which is independent of X and has mean zero. In this formula, f represents the systematic information that X provides about Y.

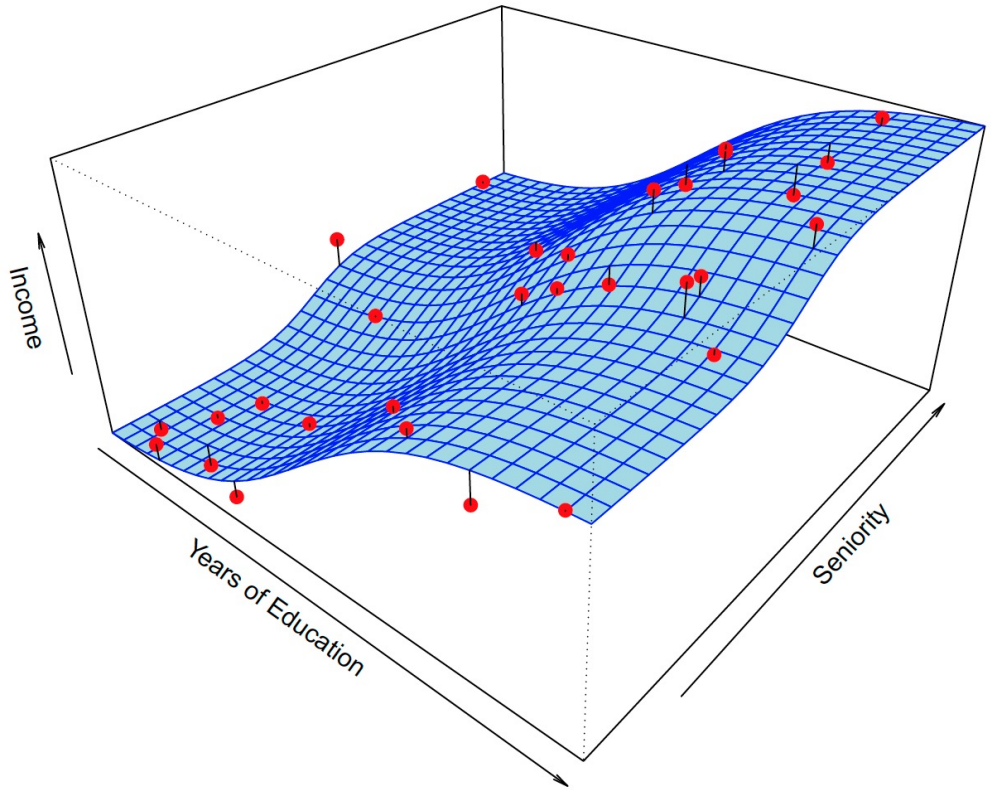
Example: Income



The blue curve represents the true underlying relationship between income and years of education, which is generally unknown (but is known in this case because the data were simulated). The black lines represent the error associated with each observation. Note that some errors are positive and some are negative. Overall, these errors have approximately mean zero.

Model

In general, the function f may involve more than one input variable. For example, we can plot *income* as a function of *years of education* and *seniority*. Here f is a two-dimensional surface that must be estimated based on the observed data.



Prediction

There are two main reasons that we may wish to estimate (know) f: *prediction* and *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

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

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.

Prediction Performance

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*.

Reducible 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 machine learning technique to estimate f.

Irreducible Error

Even we know perfectly of f, i.e., $\widehat{Y} = f(x)$, our prediction would still have some error in it! This is because Y is also a function of ϵ. Variability associated with ϵ also affects the accuracy of our predictions. This is known as the irreducible error, because no matter how well we estimate f, we cannot reduce the error introduced by ϵ.

Accuracy of Prediction

$$E[(Y-\widehat{Y})^2|X] = E\Big[(f(X)-\hat{f}(X)+\epsilon)^2|X\Big] = [f(X)-\hat{f}(X)]^2 + Var(\epsilon)$$

- Reducible error: $[f(X)-\hat{f}(X)]^2$
- Irreducible error: $Var(\epsilon)$

The focus of this course is on techniques for estimating f with the aim of minimizing the reducible error. It is important to keep in mind that the irreducible error will always provide an upper bound on the accuracy of our prediction for Y. This bound is almost always unknown in practice.

Inference

We are often interested in understanding the association between Y and $X_1, \cdots, X_p$. In this situation we wish to estimate f, but our goal is not necessarily to make predictions for Y. Now $\hat{f}$ cannot be treated as a black box, because we need to know its exact form.

Inference

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?

In this course, we will see a number of examples that fall into the prediction setting, the inference setting, or a combination of the two.

Examples

Direct Marketing

Consider a company that is interested in conducting a direct-marketing campaign. The goal is to identify individuals who are likely to respond positively to a mailing, based on observations of demographic variables measured on each individual. The demographic variables serve as predictors, and response to the marketing campaign serves as the outcome. The company is not interested in obtaining a deep understanding of the relationships; instead, the company simply wants to accurately predict the response using the predictors.

Advertising

One may be interested in answering questions such as:

- Which media are associated with sales?
- Which media generate the biggest boost in sales?
- How large of an increase in sales is associated with a given increase in TV advertising?

Real Estate

In a real estate setting, one may seek to relate values of homes to inputs such as crime rate, zoning, distance from a river, air quality, schools, income level of community, size of houses, and so forth. In this case one might be interested in the association between each individual input variable and housing price—for instance, how much extra will a house be worth if it has a view of the river? This is an inference problem. Alternatively, one may simply be interested in predicting the value of a home given its characteristics: is this house under- or over-valued? This is a prediction problem.

Estimation

Now we know that the key to understand the data, or the relationship, is to get a suitable function $\hat{f}$. In the data analysis, there are typically two types of estimation methods:

- Parametric Estimation: make an assumption about the functional form, or shape, of f. For example:

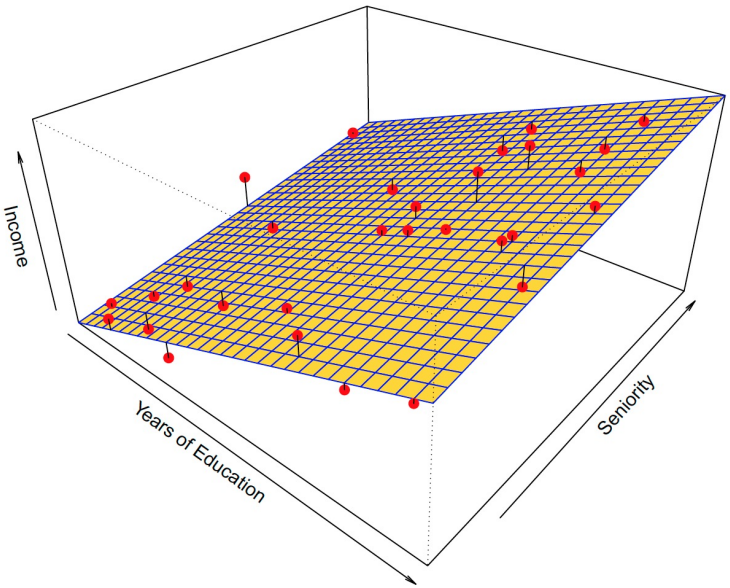
  $$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

  The procedure is now to assign/estimate $\beta = (\beta_0, \beta_1, \cdots, \beta_p)'$. Ordinary Least Square is one of popular estimation method.

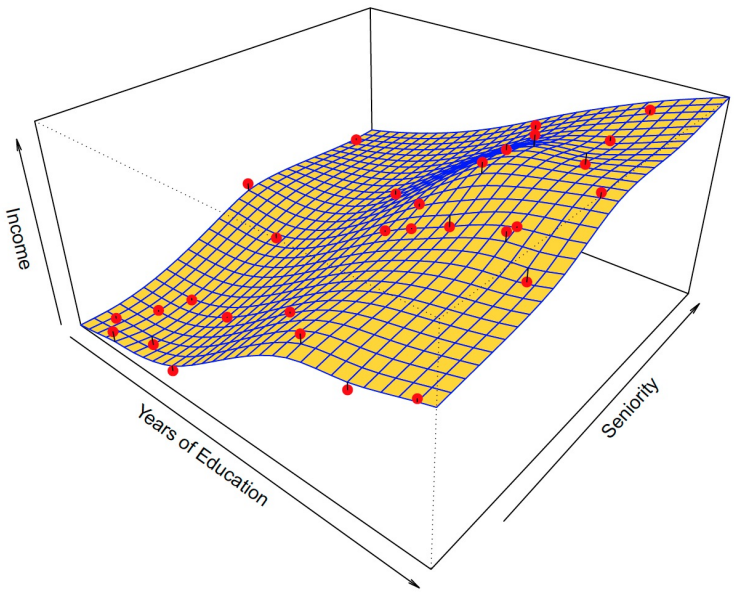
- Non-parametric Estimation: do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.

Parametric vs Non-parametric Estimation

Income–Years of Education and Seniority

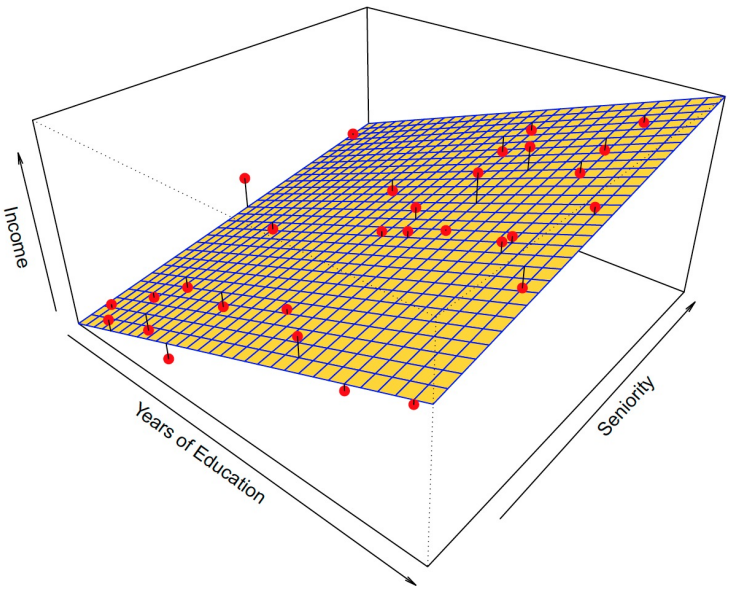


(a) Parametric

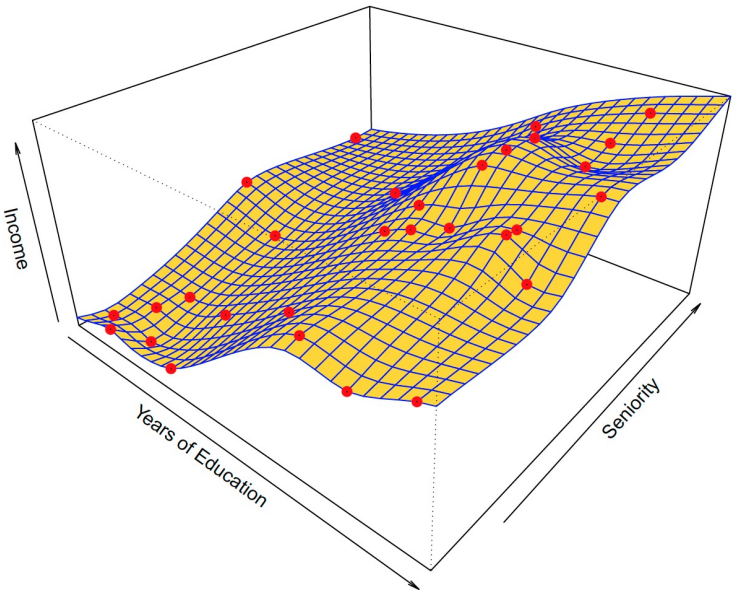


(b) Non-parametric

Income–Years of Education and Seniority



(c) Parametric



(d) Non-parametric

Accuracy and Interpretability

Of the many methods, some are less flexible, or more restrictive, in the sense that they can produce just a relatively small range of shapes to estimate f. One might reasonably ask the following question: *why would we ever choose to use a more restrictive method instead of a very flexible approach?*

- For inference restrictive models are much more interpretable, e.g., linear regression.
- In some settings, however, we are only interested in prediction, and the interpretability of the predictive model is simply not of interest.
- When interpretability is not a concern, we might expect that it will be best to use the most flexible model available. Surprisingly, this is not always the case!

Assessing Accuracy

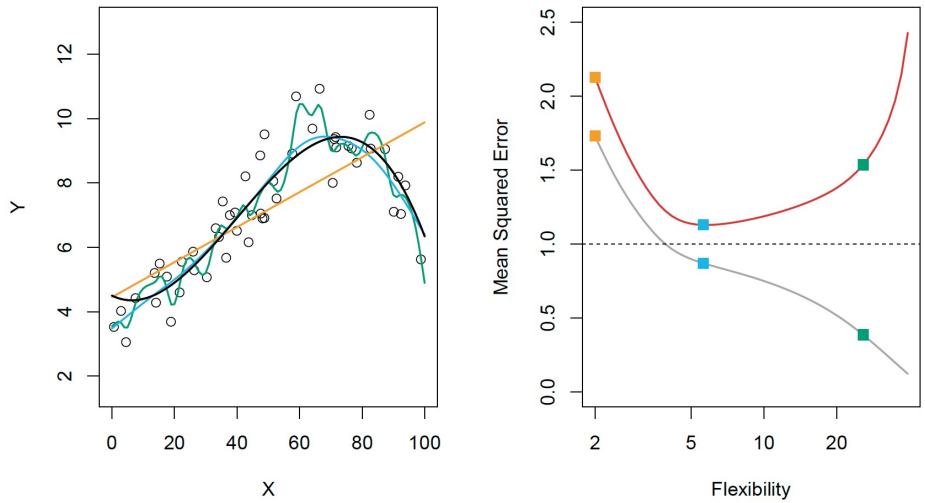
In order to evaluate the performance of a machine learning method on a given data set, we need some way to measure how well its predictions actually match the observed data. That is, we need to quantify the extent to which the predicted response value for a given observation is close to the true response value for that observation. Suppose we have observations $(y_i, x_i)_{i=1}^n$ (Training data). In the regression setting, the most commonly-used measure is the mean squared error (MSE), given by

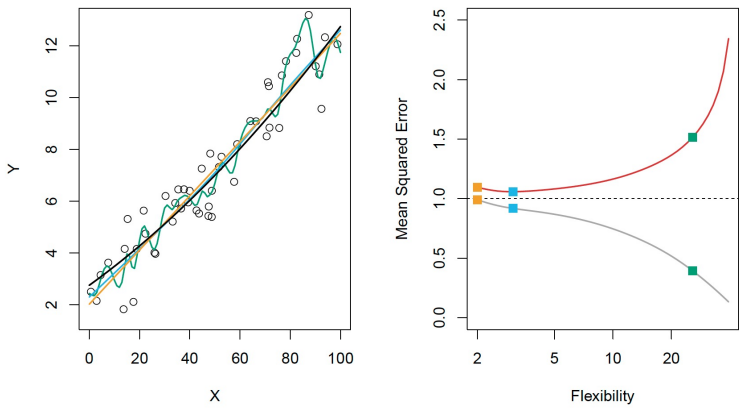
$$MSE = \sum_{i=1}^{n} [y_i - \hat{f}(x_i)]^2.$$

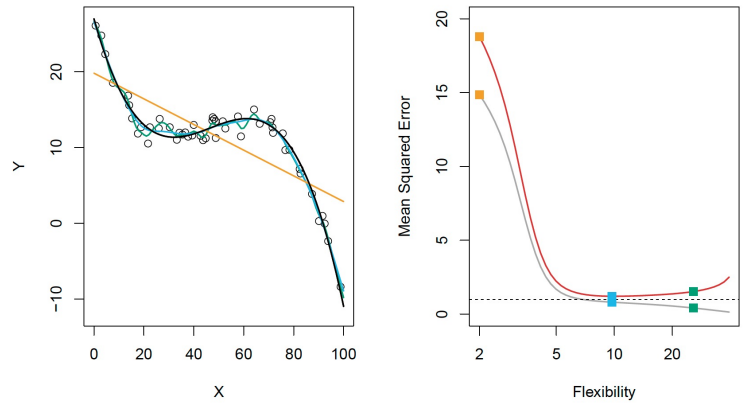
The MSE above is computed using the training data that was used to fit the model, and so should more accurately be referred to as the *training MSE*. But in general, we do not really care how well the method works training on the training data. Rather, we are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen *test data*. We call this *Test MSE*.

Overfitting

As model flexibility increases, the training MSE will decrease, but the test MSE may not. When a given method yields a small training MSE but a large test MSE, we are said to be *overfitting* the data.







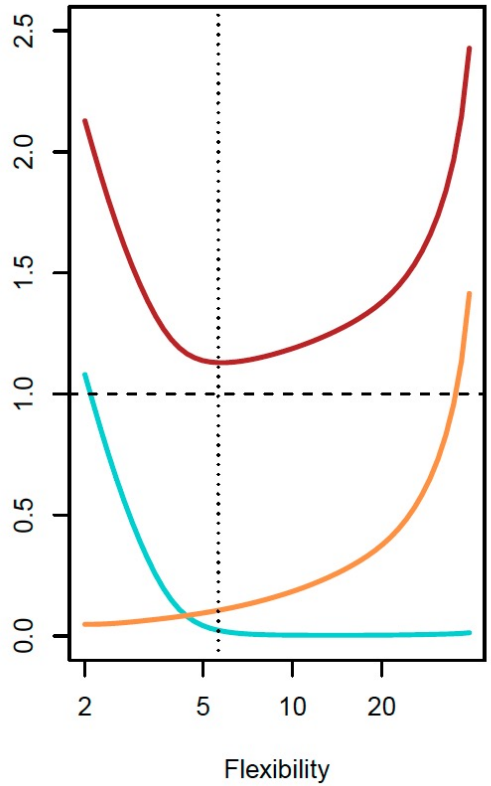
The Bias-Variance Trade-Off

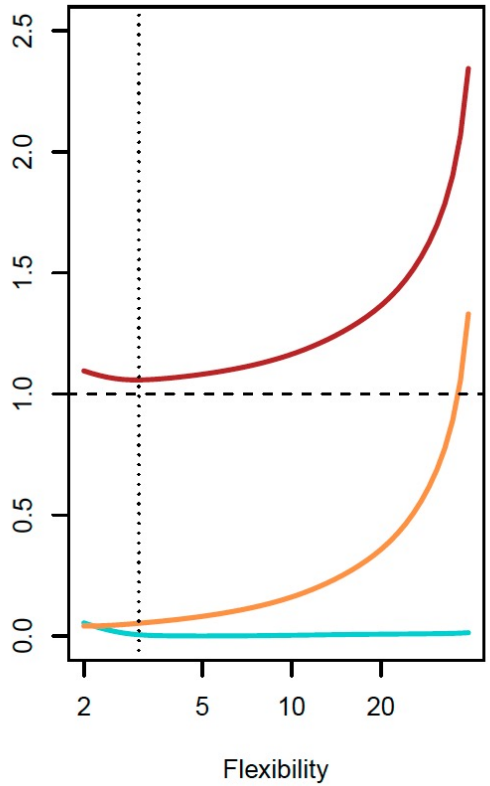
The U-shape observed in the test MSE curves turns out to be the result of two competing properties of machine learning methods. Suppose we train the data and get $\hat{f}$ using $(y_i, x_i)_{i=1}^n$. For a given value x_0, the expected test MSE, can always be decomposed into the sum of three fundamental quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$ and the variance of the error variance terms ϵ.

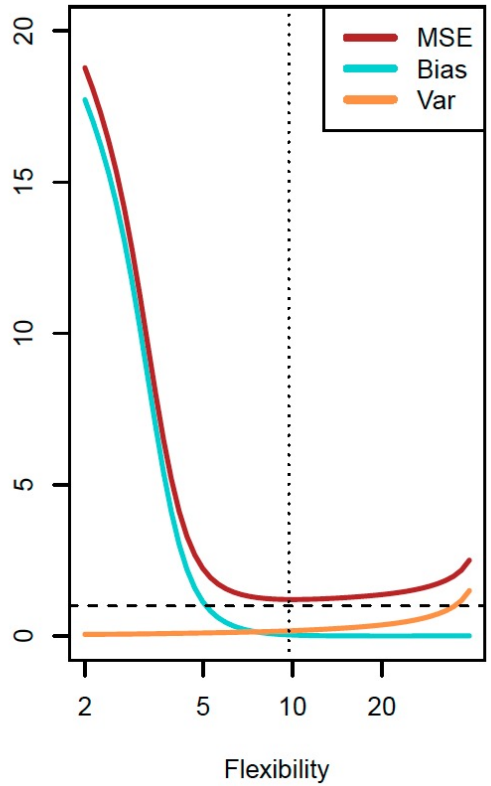
$$E[(y_0 - \hat{f}(x_0)]^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + var(\epsilon).$$

- *Variance* refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set. 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







Classification

Many of the concepts that we have encountered, such as the bias-variance trade-off, transfer over to the classification setting with only some modifications due to the fact that y_i is no longer quantitative.
Suppose that we seek to estimate f on the basis of training observations $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, where now $y_1, \cdots, y_n$ are qualitative. The most common approach for quantifying the accuracy of our estimate $\hat{f}$ is the training *error rate*, the proportion of mistakes that are made if we apply our estimate $\hat{f}$ to the training observations:

$$\frac{1}{n}\sum_{i=1}^{n} 1\{y_i = \hat{y}_i\}.$$

The test error rate associated with a set of test observations of the form test error (x_0, y_0) is given by

$$Ave\Big[1\{y_0 \neq \hat{y}_0\}\Big]. \tag{1}$$

A good classifier is one for which the test error Equation (1) is smallest.

Bayes Classifier

The test error rate given in Equation (1) is minimized, on average, by a very simple classifier that assigns each observation to the most likely class, given its predictor values. In other words, we should simply assign a test observation with predictor vector x_0 to the class j for which

$$P(Y = j|X = x_0)$$

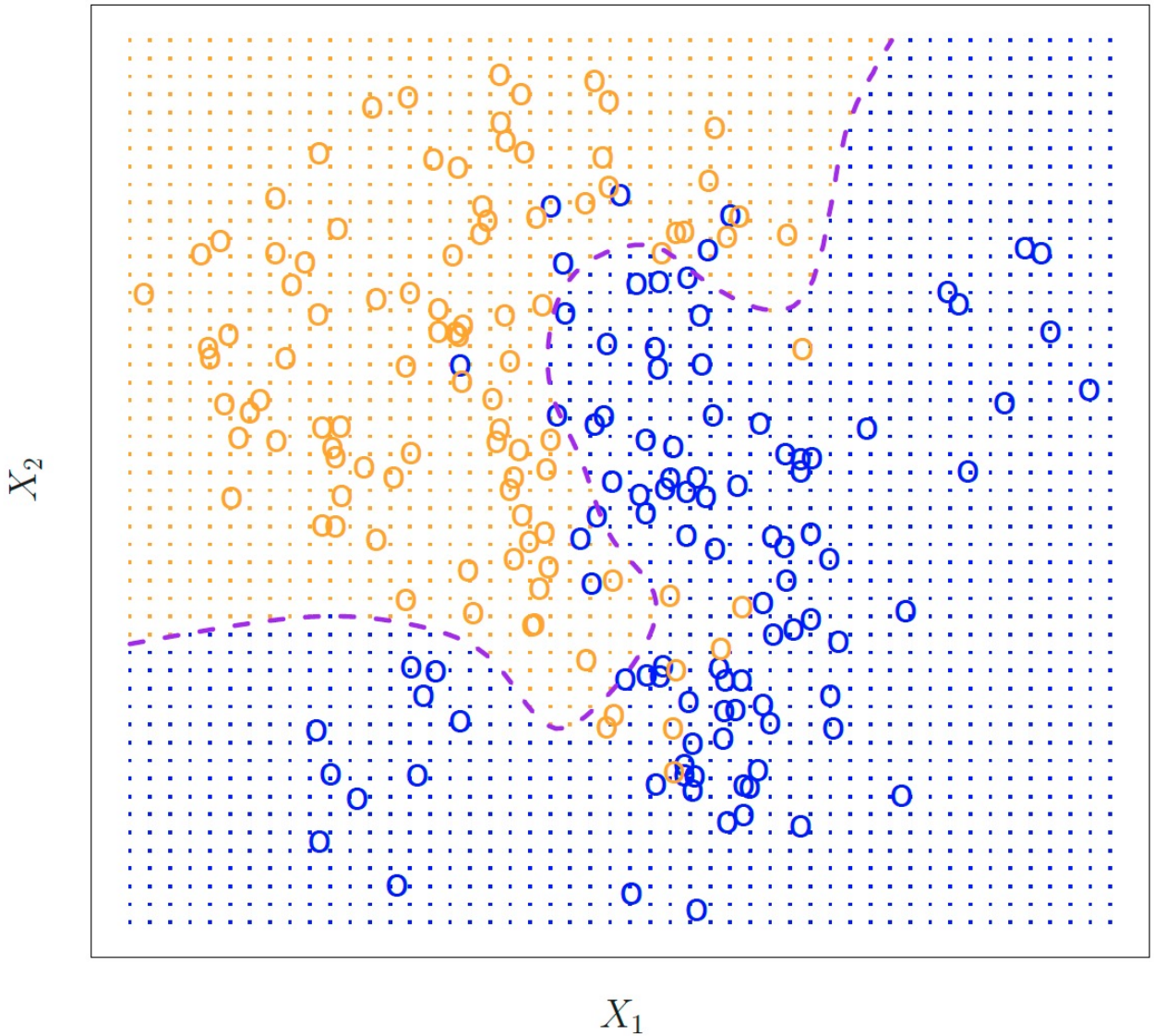
is the largest. This very simple classifier is called the *Bayes classifier*.
In the binary case, with $Y \in \{0, 1\}$, we will assign $\widehat{Y} = 1$ if $P(Y = 1|X = x_0) > 0.5$, i.e., $P(Y = 1|X = x_0) > P(Y = 0|X = x_0)$ and vice versa. The Bayes classifier produces the lowest possible test error rate, called the *Bayes error rate*. In general, the overall Bayes error rate is given by

$$1 - E\Big(\max_j P(Y = j|X = x_0)\Big).$$

The Bayes error rate is analogous to the irreducible error.

Bayes Decision Boundary

Consider an example using a simulated data set in a two dimensional space consisting of predictors X_1 and X_2.



K-Nearest Neighbor

In theory we would always like to predict qualitative responses using the Bayes classifier. But for real data, we do not know the conditional distribution of Y given X, and so computing the Bayes classifier is impossible. Therefore, the Bayes classifier serves as an unattainable gold standard against which to compare other methods. Many approaches attempt to estimate the conditional distribution of Y given X, and then classify a given observation to the class with highest estimated probability. One such method is the *K-nearest neighbors (KNN)* classifier.

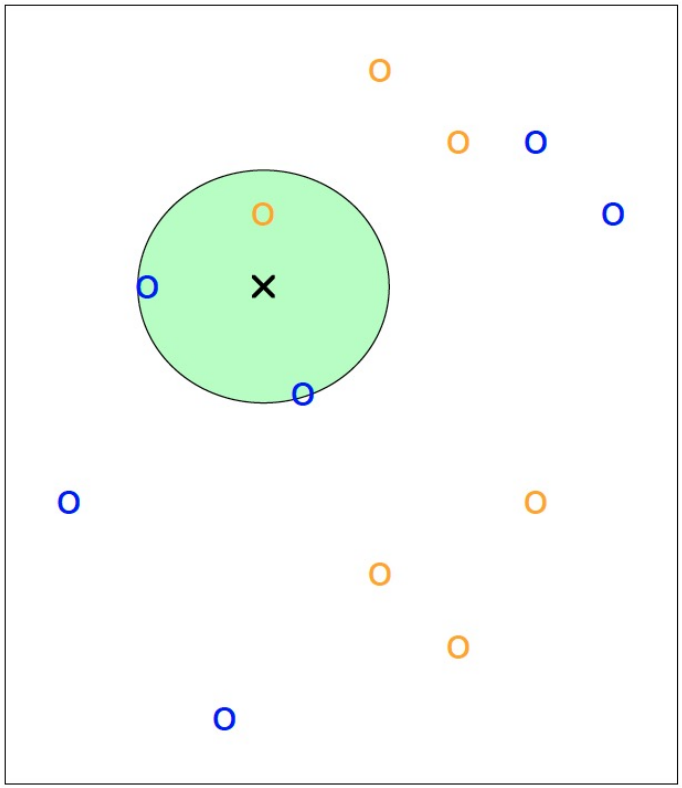
KNN

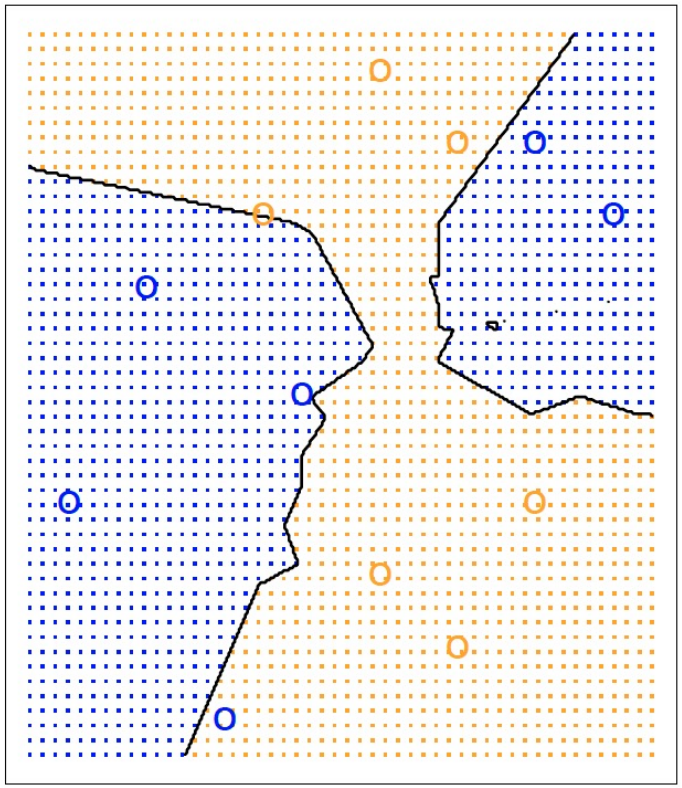
Given a positive integer K and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by $\mathcal{N}_0$. It then estimates the conditional probability for class j as the fraction of points in $\mathcal{N}_0$ whose response values equal j:

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} 1\{y_i = j\}. \tag{2}$$

Finally, KNN classifies the test observation x0 to the class with the largest probability from Equation (2).

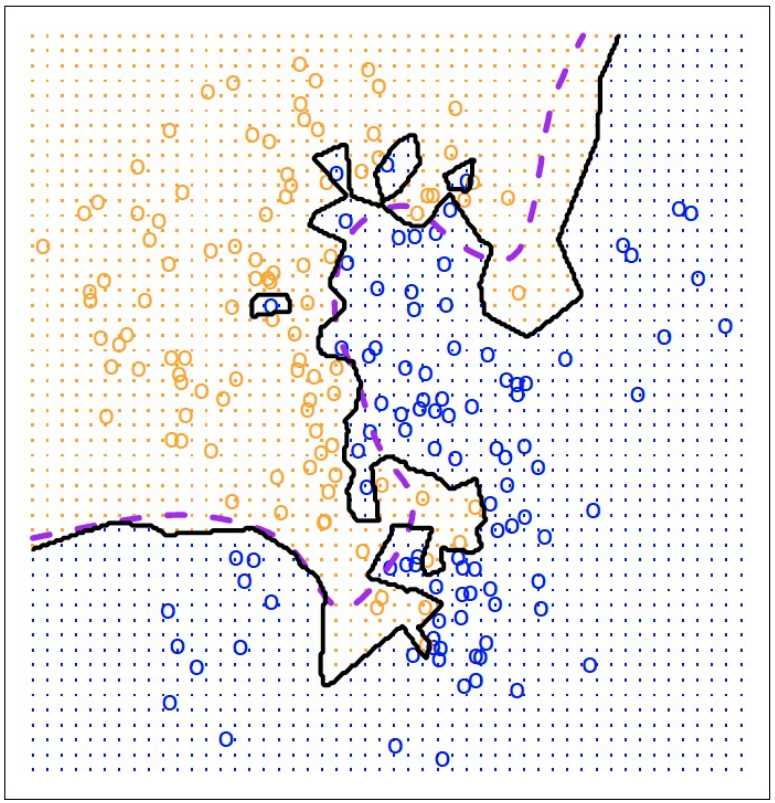
KNN





K in KNN

KNN: K=1



KNN: K=100