Leo Breiman, 2001, The Two Cultures

Leo Breiman. Statistical Learning: The Two Cultures. Statistical Science. 16 (3): 199-231, 2001.

- There are two cultures in the use of statistical modeling to reach conclusions from data.
- One assumes that the data are generated by a given stochastic data model (*The Data Modeling Culture*).
- The other uses algorithmic models and treats the data mechanism as unknown (*The Algorithmic Modeling Culture*).
- The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.

Machine Learning

- Another denominations: machine learning, statistical learning, artificial intelligence
- The techniques of Statistical Learning can help solve the problems that frequently arise when modeling an ecological problem, economic phenomenon, medical situation, climatic situation, etc..
- Idea: from a (training) data set, build and train a mathematical model f that will allow, given a new observation, to predict the category to which it belongs or some relevant output value. Predictor f is construct generally without any assumption on distribution or on nature of the dataset.
- If Y is the response:

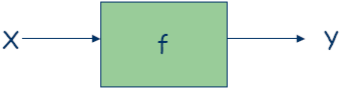
$$\text{modelisation: } Y = f(X) + \epsilon$$

$$\text{prediction: } \hat{Y} = \hat{f}(X)$$

$$\text{we want: } \hat{Y} \approx Y$$

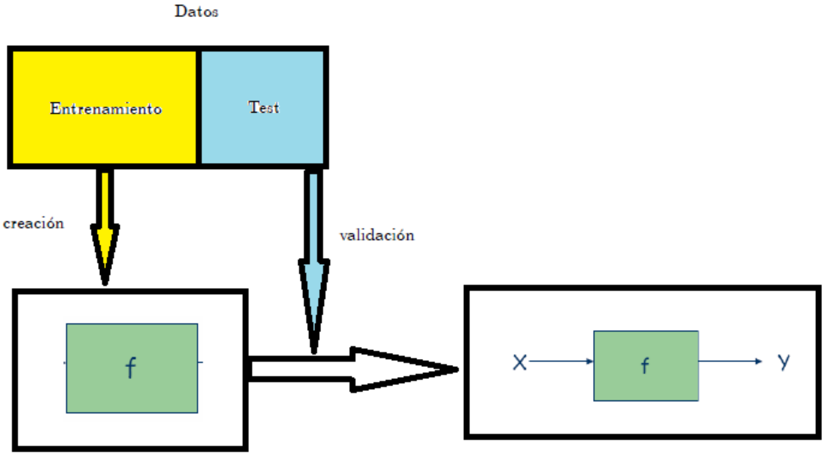
  - Data Modeling Culture: f has a given form (linear or logistic regression) and we estimate parameters from the data. Work for the model. Validation is (generally) about goodness of fit.
  - Algorithmic Modeling Culture: f is an algorithm. Validation is measured by predictive accuracy.

Examples



- Predict whether an email is spam or not spam.
- Predict whether a patient is prone to heart disease.
- Estimate the ozone rate in a city taking into account climatic variables.
- Predict the absence or presence of a species in a given environment.
- Predicting customer leaks for a financial institution.
- Identify handwritten figures of postcards in envelopes.
- Split a population into several subgroups.

Statistical Learning



Framework of Machine Learning

General framework:
$\mathcal{L}$ a data basis.

Framework of Machine Learning

General framework:
$\mathcal{L}$ a data basis. We search about $f : \mathcal{X} \to \mathcal{Y}$ a good predictor or a good explainer.

- Supervised Learning: $\mathcal{L} = \{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$

  X: input variable, independent variable, explanatory (real o multidimensional), continuous, categorical, binary, ordinal.

  Y: output variable, dependent variable, real o categorical.

  - Classification: $y \in \{-1, 1\}$ (binary) or $y \in \{1, \dots, K\}$ (multiclass).
  - Regression: $y \in \mathbb{R}$.

- Unsupervised Learning $\mathcal{L} = \{x_1, \dots, x_n\} \subset \mathcal{X} \subset \mathbb{R}^d$

  - Clustering
  - Density estimation

  In all cases, the sample $\mathcal{L}$ is a collection of n independents realization of a multivariate random variable (X, Y) or X

When you are working with data...

and you are thinking about a model to use, it is useful to remember that:

- Data Modeling Culture - Algorithm Modeling Culture
- Supervised - Unsupervised
- Supervised: Clasification - Regression
- Types of variables, Missing Values
- Accuracy of the method is important but it is even more important over *test data*.
- Multiplicity of good models: aggregation methods
- Occam's razor dilemna: simpler is better? Simplicity vs Accuracy?
- Curse of Dimensionality? Handicap or blessing?
- *(The focus)..is on solving the problem instead of asking what data model (they can create). The best solution could be an algorithmic model, or may be a data model, or may be a combination. But the trick to being a scientist is to be open to using a wide variety of tools*, Breiman, The two cultures.
- *All models are wrong, but some are useful*, Georges Box (1919-2013)

Data matrix

Two ways to consider the data matrix $\mathbf{X} = ((x_{ij}))_{i=1,\dots,n}^{j=1,\dots,p} \in \mathcal{M}_{n\times p}$ (n observations with p variables).

By rows (observations):

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} \text{obs 1} \\ \text{obs 2} \\ \vdots \\ \text{obs n} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}$$

By columns (variables):

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} v & v & & v \\ a & a & & a \\ r & r & & r \\ i & i & \dots & i \\ a & a & & a \\ b & b & & b \\ l & l & & l \\ e & e & & e \\ 1 & 2 & & p \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \dots & x_p \end{pmatrix}$$

Let y_i the response of observation i. Our data set is :

$$\mathcal{L} = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \dots, (\mathbf{x_n}, y_n)\}$$

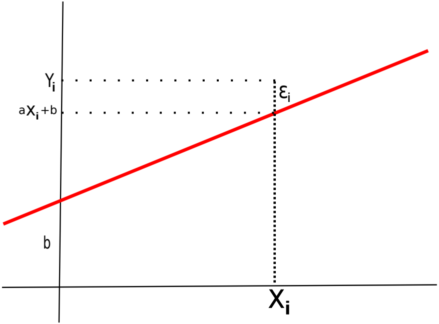
where $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \dots, (\mathbf{x_n}, y_n)$ are independent realizations of variable (X, Y) where Y is dependent of X.

Simple Linear Model: method of least squares

Data: $\mathcal{L} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Simple Linear Model: method of least squares

Data: $\mathcal{L} = \{(x_1, y_1), \dots, (x_n, y_n)\}$.
We look for the line $y = ax + b$ that passes as close as possible to the data.



We find a and b that minimize the sum of squared errors

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)^2$$

The simple linear regression model is

$$y_i = \underbrace{ax_i + b}_{y_{\text{est}}} + \varepsilon_i, \; \forall\, i = 1, \dots, n$$

Linear Model: method of least squares

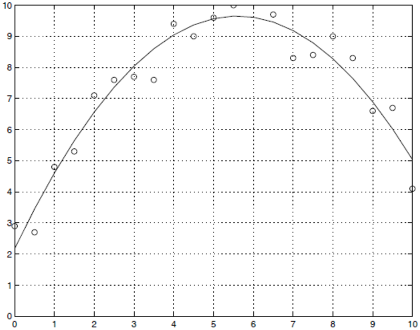
The above method can be easily extended.

Linear Model: method of least squares

The above method can be easily extended.
For example the parabola that adjusts a set of points:

Linear Model: method of least squares

The above method can be easily extended.
For example the parabola that adjusts a set of points:



$$y = a + bx + cx^2$$

(linear model on the coefficients!)

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix}$$

Multiple Linear Regression

Now we want to predicto a real random variable $Y \in \mathbb{R}$ from d real variables $X_1, \dots, X_d$. We consider model:

$$f(X) = \beta_0 + \beta_1 X_1 + \dots + \beta_d X_d$$

As in simple linear regression, if $\mathcal{L} = \{(\mathbf{x_1}, y_1,), \dots, (\mathbf{x_n}, y_n)\}$ is the data set, we look at a vector $\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix} \in \mathbb{R}^{d+1}$ that minimizes

$$\sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_d x_{id})\right)^2$$

Observe that $\sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_d x_{id})\right)^2 = ||Y - X\beta||^2$ so we have a linear algebra problem:

$$X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{nd} \end{pmatrix}_{n \times (d+1)}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix} \in \mathbb{R}^{d+1}$$

whose solution is given by $(X^t X)\beta = X^t \mathbf{y}$.

Classification and Regression Trees (CART)

Classification **A**nd **R**egression **T**rees (Breiman 1984).

Two types of trees: regression trees to predict continuous variables and classification trees to predict categorical variables.

The tree is constructed from binary partitions with respect to the coordinates of the data. For example if the variables are $X_1, \ldots, X_d$, the cut condition for the data will be of type $X_2 < c$ or $X_2 \geq c$ if X_2 is continuous or $X_2 \in \mathcal{A}$ or $X_2 \notin \mathcal{A}$ if X_2 is categorical.

Three steps:

1. Binary separation of the data of each node in two subnodes according to some criterion;

Classification and Regression Trees (CART)

Classification **A**nd **R**egression **T**rees (Breiman 1984).

Two types of trees: regression trees to predict continuous variables and classification trees to predict categorical variables.

The tree is constructed from binary partitions with respect to the coordinates of the data. For example if the variables are $X_1, \ldots, X_d$, the cut condition for the data will be of type $X_2 < c$ or $X_2 \geq c$ if X_2 is continuous or $X_2 \in \mathcal{A}$ or $X_2 \notin \mathcal{A}$ if X_2 is categorical.

Three steps:

1. Binary separation of the data of each node in two subnodes according to some criterion;
2. Decision of the size of the tree: stop and prune criteria

Classification and Regression Trees (CART)

Classification **A**nd **R**egression **T**rees (Breiman 1984).

Two types of trees: regression trees to predict continuous variables and classification trees to predict categorical variables.

The tree is constructed from binary partitions with respect to the coordinates of the data. For example if the variables are $X_1, \ldots, X_d$, the cut condition for the data will be of type $X_2 < c$ or $X_2 \geq c$ if X_2 is continuous or $X_2 \in \mathcal{A}$ or $X_2 \notin \mathcal{A}$ if X_2 is categorical.

Three steps:

1. Binary separation of the data of each node in two subnodes according to some criterion;
2. Decision of the size of the tree: stop and prune criteria
3. Assigning a class or value to terminal nodes.

CART

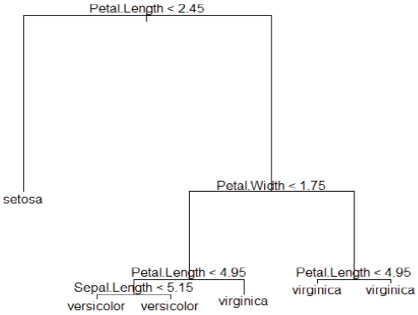
Example: Iris

Goal: Predict the species of the iris flower.

Data: 150 flowers

Dependent variable: Species (setosa, virginica, versicolor)

Independents variables: Sepal Length, Petal Length, Sepal Width, Petal Width



CART

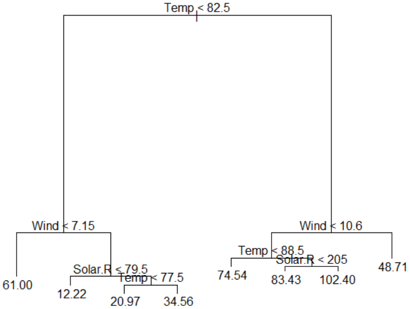
Example: airquality

Goal: Predict the ozone level in New York.

Data: 153 days

Dependent Variable: ozone level

Independents variables Date, Solar Radiation, Wind and Temperature



CART

Easy to interpret, but ... very unstable: a small change in the sample leads to completely different results.

CART

Easy to interpret, but ... very unstable: a small change in the sample leads to completely different results.

Aggregation Methods:

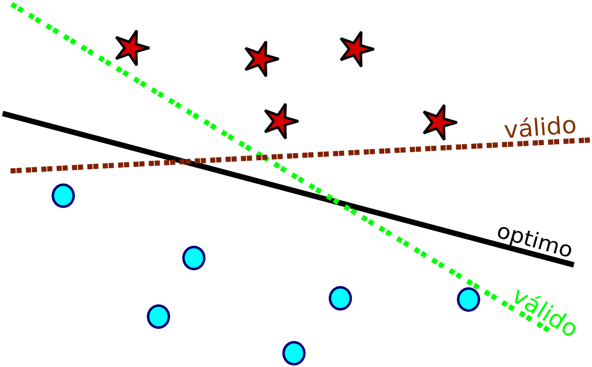
1. **Bagging** (Breiman, 1996): average of several trees based on data re-samples.
2. **Random Forests** (Breiman, 2001): combines the Bagging and CART algorithms.
3. **Boosting** (Freund and Shapire, 1997): weighted average of trees. The weighting takes into account the performance of each tree in each stage of the algorithm.

Support Vector Machines (SVM)

In the classification context, SVM (Vapnik, 1995) is a method that consists of finding a curve that separates the data as best as possible.

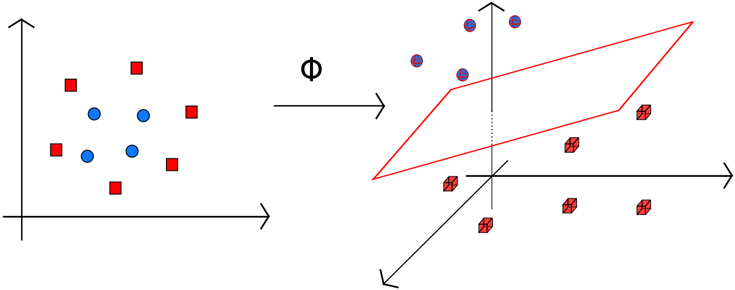
Support Vector Machines (SVM)

In the classification context, SVM (Vapnik, 1995) is a method that consists of finding a curve that separates the data as best as possible.
If the data are linearly separable:



Support Vector Machines

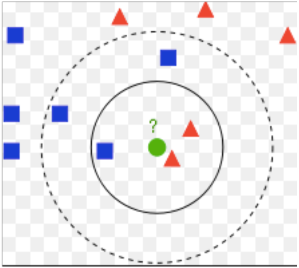
If the data are not linearly separable, we transform them to a space where they are:



k-Nearest Neighbor (*k*-NN)

In *k*-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

In *k*-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.



A little formality

A little formality

Consider a *loss function* L, i.e $L(y, u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

A little formality

Consider a *loss function* L, i.e $L(y, u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

Ejemplos:

1. $L(y, u) = 1_{\{y \neq u\}}$ (classification)

A little formality

Consider a *loss function* L, i.e $L(y, u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

Ejemplos:

1. $L(y, u) = 1_{\{y \neq u\}}$ (classification)
2. $L(y, u) = (y - u)^2$ (regression)

A little formality

Consider a *loss function* L, i.e $L(y, u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

Ejemplos:

1. $L(y, u) = 1_{\{y \neq u\}}$ (classification)
2. $L(y, u) = (y - u)^2$ (regression)
3. $L(u) = -log(u)$ (density estimation)

A little formality

Consider a *loss function* L, i.e $L(y,u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

Ejemplos:

1. $L(y,u) = 1_{\{y \neq u\}}$ (classification)
2. $L(y,u) = (y-u)^2$ (regression)
3. $L(u) = -log(u)$ (density estimation)

We look for a function f_C (the original), among all the functions of a certain class $\mathcal{C}$, that minimizes the expected value of L (which we call *risk* or *Expected Predictive Error*), i.e:

$$f_C = \underset{f\in\mathcal{C}}{\text{Argmin}} R_L(f) = \underset{f\in\mathcal{C}}{\text{Argmin}} \mathbb{E}\big(L(Y,f(X))\big)$$

A little formality

Consider a *loss function* L, i.e $L(y, u)$ which measures the cost of deciding $u = f(x)$ for the input x knowing that y is the true output.

Ejemplos:

1. $L(y, u) = 1_{\{y \neq u\}}$ (classification)
2. $L(y, u) = (y - u)^2$ (regression)
3. $L(u) = -log(u)$ (density estimation)

We look for a function f_C (the original), among all the functions of a certain class $\mathcal{C}$, that minimizes the expected value of L (which we call *risk* or *Expected Predictive Error*), i.e:

$$f_C = \underset{f \in \mathcal{C}}{\text{Argmin}} R_L(f) = \underset{f \in \mathcal{C}}{\text{Argmin}} \mathbb{E}\big(L(Y, f(X))\big)$$

The choice of $\mathcal{C}$ depends on the nature of the phenomenon being modeled, the hypotheses and experience on the data available, the opinion of the experts, etc.

Problem: It is impossible to search for such $f_{\mathcal{C}}$.

A little formality

In practice, this predictor is constructed from a data set $\mathcal{L} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i \in \mathcal{X} \subset \mathbb{R}^d$ and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$ or $y_i \in \mathcal{Y} \subset \mathbb{R}$ where it supposed that all the n labeled observations of $\mathcal{L}$ are independent realization of the variable (X, Y) with unknown distribution law.
As it is impossible to lead with the expected risk (as distribution of (X, Y) is unknown), the goal consists to minimize the empirical risk

$$R_{n,L}(f) = \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(x_i)\big)$$

That is to search a function $\widehat{f_n} \in \mathcal{C}$ such that:

$$\widehat{f_n} = \underset{f \in \mathcal{C}}{\operatorname{Argmin}} R_{n,L}(f) = \underset{f \in \mathcal{C}}{\operatorname{Argmin}} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(x_i)\big)$$

The classification problem

For example, in a classification problem if $y \in \{1, \dots, K\}$, we use as loss function $L(x, y, u) = \mathbb{1}_{\{u \neq y\}}$.

The classification problem

For example, in a classification problem if $y \in \{1, \ldots, K\}$, we use as loss function $L(x, y, u) = \mathbb{1}_{\{u \neq y\}}$.

The associated risk with L is:

$$R_L(f) = \mathbb{P}\big(Y \neq f(X)\big)$$

and the empirical risk is

$$R_{L,n}(f) = \frac{1}{n} \#\{i : f(x_i) \neq y_i\}$$

The function that minimizes $R_L(f)$ is

$$f^*(x) = \underset{k \in \{1,\ldots,K\}}{\text{Argmax}} \; \mathbb{P}(Y = k | X = x)$$

and predicts the class k that maximizes the posterior probability of Y knowing X. This classifier is known as *Bayes classifier* and can be interpreted as follows:, the problem is reduced in looking for that function that minimizes the amount of errors committed on the sample.

The regression problem

In a regression problem we look at a function $f : \mathbb{R}^d \to \mathbb{R}$ so that, for a new observation (x, y), the prediction $f(x)$ is a good approximation of y in the sense that distance between $f(x)$ and y is small. We use as loss function $L(y, u) = (u - y)^2$.
the associate risk L is:

$$R_L(f) = \mathbb{E}_{(X,Y)}\left[(Y - f(X))^2\right]$$

and the empirical risk is

$$R_{L,n}(f) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f(x_i)\big)^2$$

The function that minimizes $R_L(f)$ is

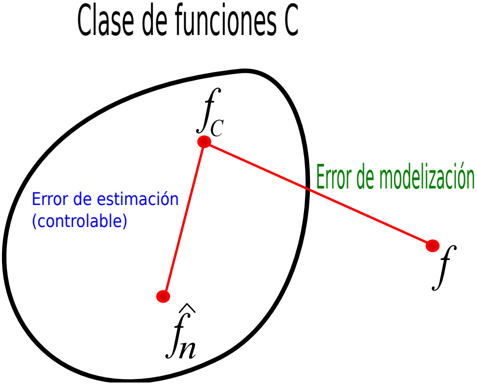
$$f^*(x) = m(x) = \mathbb{E}(Y|X = x)$$

If instead of minimizing theoretical risk we minimize empirical risk, then the solution is the function that minimizes the least squares method.

Approach errors

Let summarize the different functions previously encountered:

- f is the theoretical predictor (we don't know it).
- f_C is the best among all possible predictors within a class of functions $\mathcal{C}$ (we don't know it).
- $\hat{f}_n$ is the predictor we use in practice, the function that minimizes empirical risk:



Approach errors

- Modelling error (associated with bias): $f - f_C$
  It depends on the choice of class $\mathcal{C}$. Observe that if we consider as the family of all possible functions, we will have overfitting.
- Estimation error (associated with the variance): $\hat{f}_n - f_C$
  It is a statistical error, if the size of the sample is large, under certain hypotheses about the class $\mathcal{C}$, it is true that $\hat{f}_n$ converge, when n tends to infinity to f_C. In fact it is a convergence of the risks (Vapnik's theorem)

Approach errors

- Modelling error (associated with bias): $f - f_C$
  It depends on the choice of class $\mathcal{C}$. Observe that if we consider as the family of all possible functions, we will have overfitting.
- Estimation error (associated with the variance): $\hat{f}_n - f_C$
  It is a statistical error, if the size of the sample is large, under certain hypotheses about the class $\mathcal{C}$, it is true that $\hat{f}_n$ converge, when n tends to infinity to f_C. In fact it is a convergence of the risks (Vapnik's theorem)

Theorem 1

The Fundamental Theorem of Learning (Vapnik, 1997) states that, under certain conditions on the class of functions $\mathcal{C}$, $\hat{f}_n$ "converges" to f_C (risks through) . These conditions are related to the dimension of Vapnik-Chervonenkis (VC dimension) of the function class $\mathcal{C}$. The VC dimension measures "how big" is an infinite class of functions, so if $\mathcal{C}$ is not too large, that is, the VC dimension is finite, is in the hypothesis of the Fundamental Theorem of Learning

How estimate f?

The goal is from a sample $\mathcal{L} = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \dots, (\mathbf{x_n}, y_n)\}$ estimate an unknown function f, finding an estimator $\widehat{f}$ such that

$$y \approx \widehat{f}(x)$$

for a new observation (x, y). As we say before, we suppose that observations of $\mathcal{L}$ are n independent realizations of a multivariate random variable (X, Y) of unknown distribution.

1. *Parametric methods*. The problem of estimating f is reduced to estimate some parameters, after assuming that f belongs to a certain family of functions.

   1) An assumption is made about the shape of the model, for example linear

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

   where we have to estimate $\beta_0, \beta_1, \dots, \beta_p$.

   1. After the model is selected, it is trained from $\mathcal{L}$. For example, in the case of the linear model,

   $$\widehat{\beta} = (X'X)^{-1}X'Y$$

   where

   $$X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}_{n \times (p+1)}, \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \ \widehat{\beta} = \begin{pmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \\ \vdots \\ \widehat{\beta}_p \end{pmatrix} \in \mathbb{R}^{p+1}$$

How estimate f?

2) *Non parametric methods*. No assumption is made about the nature of f. In general, it allows covering a greater spectrum of forms for f, making the model more plausible to the true f. However, in general, a large number of observations is needed to obtain a performant model.

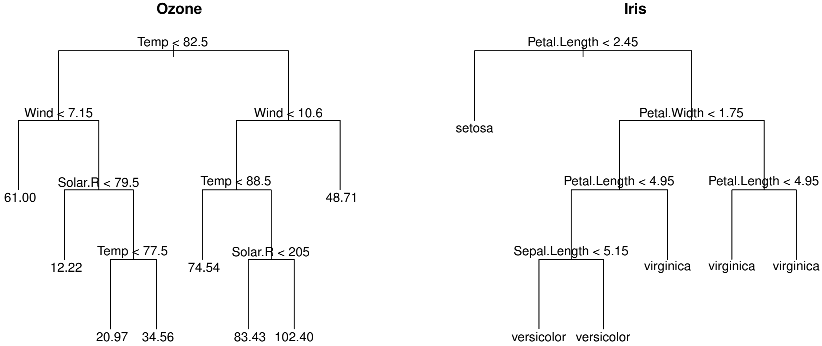
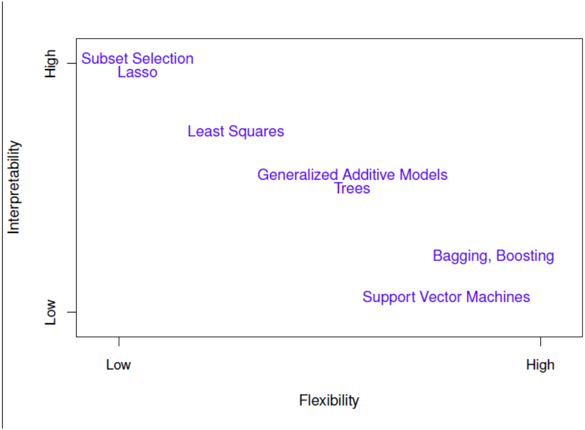


Figure: Classification and Regression Trees (Breiman, 1984)

Performance vs Interpretability



Evaluation of the model

1. In regression quality of the fitting of a predictor can be evaluated by the *mean squared error MSE*:

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

It will be small if the predictions are close to the true response values and large if for some observations the prediction and the label are very different.

Evaluation of the model

1. In regression quality of the fitting of a predictor can be evaluated by the *mean squared error MSE*:

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

   It will be small if the predictions are close to the true response values and large if for some observations the prediction and the label are very different.

   However, evaluating the performance of the model on the data with which it has been trained, is not very interesting, or at least it is not as interesting as evaluating it on fresh data, which were not used for the estimation of $\widehat{f}$.

Evaluation of the model

1. In regression quality of the fitting of a predictor can be evaluated by the *mean squared error MSE*:

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

It will be small if the predictions are close to the true response values and large if for some observations the prediction and the label are very different.

However, evaluating the performance of the model on the data with which it has been trained, is not very interesting, or at least it is not as interesting as evaluating it on fresh data, which were not used for the estimation of $\widehat{f}$.

The performance of $\widehat{f}$ (construct over $\mathcal{L}$) is evaluated on a *testing set* $\mathcal{T} = \{(\mathbf{z_1}, u_1), (\mathbf{z_2}, u_2), \dots, (\mathbf{z_s}, u_s)\}$ computing the *test*-MSE (generalization error):

$$\frac{1}{s}\sum_{i=1}^{s}\left(u_i - \widehat{f}(z_i)\right)^2$$

Evaluation of the model

1. In regression quality of the fitting of a predictor can be evaluated by the *mean squared error MSE*:

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

It will be small if the predictions are close to the true response values and large if for some observations the prediction and the label are very different.

However, evaluating the performance of the model on the data with which it has been trained, is not very interesting, or at least it is not as interesting as evaluating it on fresh data, which were not used for the estimation of $\widehat{f}$.

The performance of $\widehat{f}$ (construct over $\mathcal{L}$) is evaluated on a *testing set* $\mathcal{T} = \{(\mathbf{z_1}, u_1), (\mathbf{z_2}, u_2), \ldots, (\mathbf{z_s}, u_s)\}$ computing the *test*-MSE (generalization error):

$$\frac{1}{s}\sum_{i=1}^{s}\left(u_i - \widehat{f}(z_i)\right)^2$$

In practice, original data set is divided in two parts: the first, $\mathcal{L}$, usually 2/3, to train the model, and the remaining 1/3, $\mathcal{T}$, to test it. Also in this way, the overfitting is avoided

Evaluation of the model

1. In regression quality of the fitting of a predictor can be evaluated by the *mean squared error MSE*:

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

   It will be small if the predictions are close to the true response values and large if for some observations the prediction and the label are very different.

   However, evaluating the performance of the model on the data with which it has been trained, is not very interesting, or at least it is not as interesting as evaluating it on fresh data, which were not used for the estimation of $\widehat{f}$.

   The performance of $\widehat{f}$ (construct over $\mathcal{L}$) is evaluated on a *testing set* $\mathcal{T} = \{(\mathbf{z_1}, u_1), (\mathbf{z_2}, u_2), \dots, (\mathbf{z_s}, u_s)\}$ computing the *test*-MSE (generalization error):

   $$\frac{1}{s}\sum_{i=1}^{s}\left(u_i - \widehat{f}(z_i)\right)^2$$

   In practice, original data set is divided in two parts: the first, $\mathcal{L}$, usually $2/3$, to train the model, and the remaining $1/3$, $\mathcal{T}$, to test it. Also in this way, the overfitting is avoided

2. In classification the error is measured with the misclassified rate:

   $$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{y_i \neq \widehat{y}_i\}}$$

   where $\widehat{y}_i$ is the class prediction of f for observation i

Bias-variance trade-off

If we assume that $y = f(x) + \epsilon$, it is possible to prove that the expected value of the MSE for a fixed test value x_0, can be decomposed as:

$$\mathbb{E}\big(y_0 - \widehat{f}(x_0)\big)^2 = \mathrm{Var}\big(\widehat{f}(x_0)\big) + \big[\mathrm{Sesgo}\big(\widehat{f}(x_0)\big)\big]^2 + \mathrm{Var}(\epsilon)$$

Bias-variance trade-off

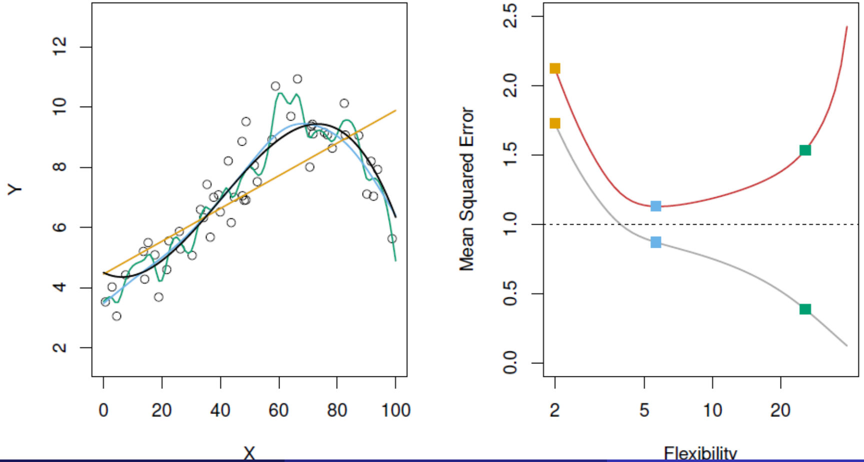
If we assume that $y = f(x) + \epsilon$, it is possible to prove that the expected value of the MSE for a fixed test value x_0, can be decomposed as:

$$\mathbb{E}\big(y_0 - \widehat{f}(x_0)\big)^2 = \mathrm{Var}\big(\widehat{f}(x_0)\big) + \big[\mathrm{Sesgo}\big(\widehat{f}(x_0)\big)\big]^2 + \mathrm{Var}(\epsilon)$$

- As $\mathrm{Var}\big(\widehat{f}(x_0)\big)$ and $\big[\mathrm{Sesgo}\big(\widehat{f}(x_0)\big)\big]^2$ are non negatives, it follows that $\mathbb{E}\big(y_0 - \widehat{f}(x_0)\big)^2$ has as lower bound $\mathrm{Var}(\epsilon)$.
- We call *variance* to the amount that varies $\widehat{f}$ if we change the training set (different set of workouts produce different $\widehat{f}$). Under ideal conditions, the estimate of f does not change much if we change the training sets. In general, very flexible statistical models (with many parameters) have high variance. For example in the case of simple linear regression, when we change an element of the data set, the estimator does not vary so much. On the other hand if the model is very adjusted, changing a point produces a significant change in the estimation.
- *Bias* refers to the modelling error: explaining a real and complicated problem by a simpler mathematical model. For example, linear models assume that there is a linear relationship between Y and explanatory variables $X_1, \dots, X_p$ which clearly has little chance of happening, so the bias will be important. In general, flexible statistical methods have a little bias.

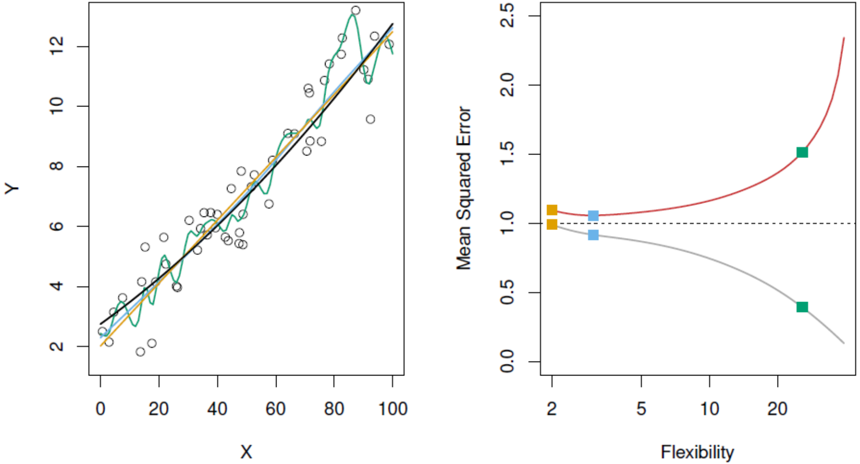
Bias-variance trade-off. Example

Several estimators (smoothing splines) are considered for different data sets (example extracted of James, Witten, Hastie and Tibshirani book).
Example 1. On the left hand three estimators with different flexibility adjusting the same data points and on the right hand the MSE curve of the flexibility on the training set (grey) and on a generalization set (red).



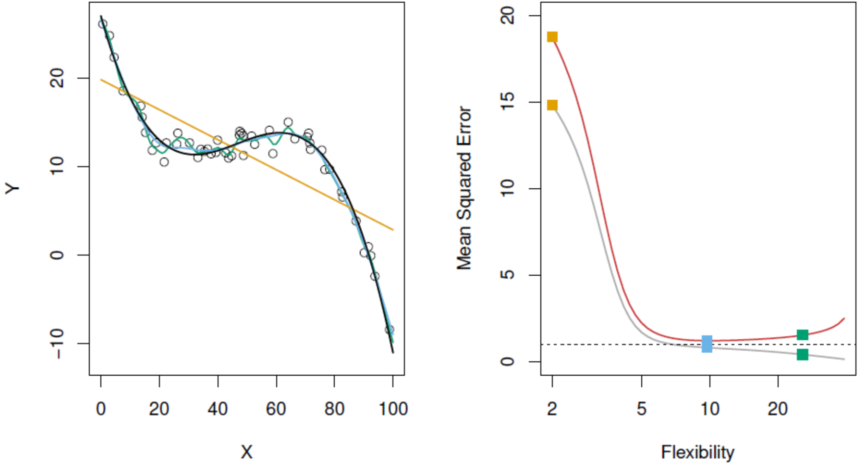
Bias-variance trade-off. Example

Example 2. On the left hand three estimators with different flexibility adjusting the same data points and on the right hand the MSE curve of the flexibility on the training set (grey) and on a generalization set (red).



Bias-variance trade-off. Example

Example 3. On the left hand three estimators with different flexibility adjusting the same data points and on the right hand the MSE curve of the flexibility on the training set (grey) and on a generalization set (red).



Bias-variance trade-off. Example

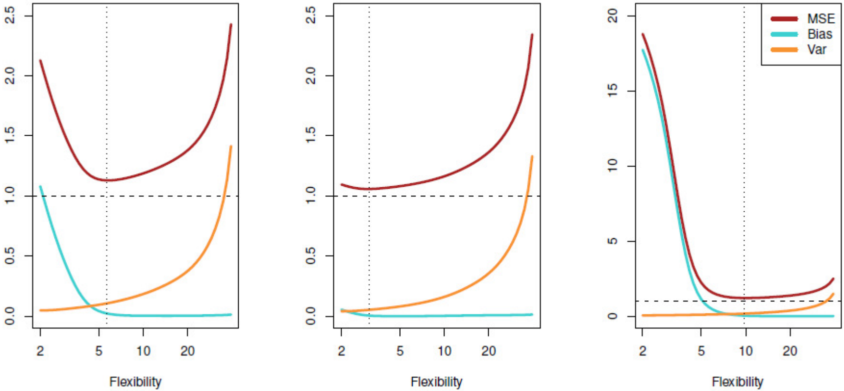


Figure: The three graphs refer to the MSE, bias and variance curves of three previous examples

Bias-variance trade-off. Example

The choice of the model will also be important to consider it a classification problem:

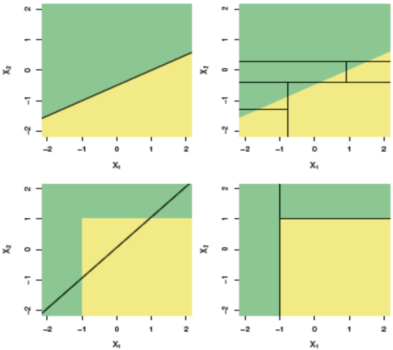


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

Referencias

- D. Peña, *Análisis de Datos Multivariantes*, Mac Graw Hill, 2002.
- A. I. Izenman, *Modern Multivariate Statistical Techniques*, Springer, 2008.
- G. James, D. Witten, T. Hastie, R.Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer Texts in Statistics, 2013
- Devroye,L., Györfi,L. and Lugosi,G. A Probability Theory of Pattern Recognition. Springer, 1996
- Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, Springer, 2001.