Introduction to Modelling and to Statistical Learning (Part 2)

Mathias Bourel

IMERL - Facultad de Ingeniería, Universidad de la República, Uruguay

March 21, 2019

Plan

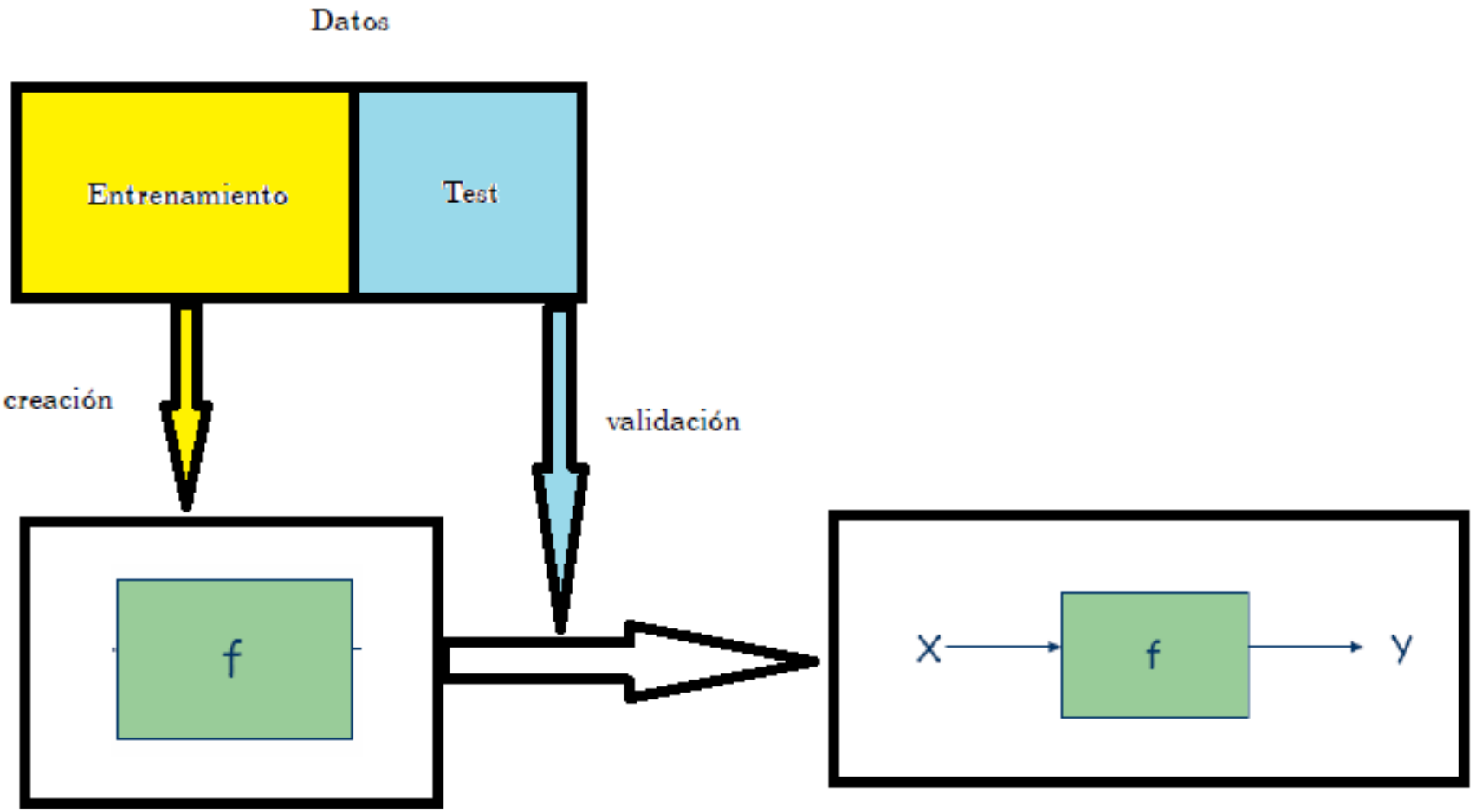
Introduction

- 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 model that will allow, given a new observation, to predict the category to which it belongs or some relevant output value.

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

  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 $y \in \{1, \ldots, K\}$ (multiclass).
  - Regression: $y \in \mathbb{R}$.

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

  - 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

A little formality

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

A little formality

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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

Let see how three famous problem in Statistics can be viewed as statistical learning problem.

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.

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:

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

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 classification problem

Suppose our problem is binary and we want to classify the observations into two categories: 0 and 1. In this case the Bayes classifier is the function f^* that minimizes the probability of being wrong:

$$f^* = \underset{f:\mathcal{X}\to\{0,1\}}{\text{Argmin}}\ \mathbb{P}(f(X) \neq Y)$$

The classification problem

Suppose our problem is binary and we want to classify the observations into two categories: 0 and 1. In this case the Bayes classifier is the function f^* that minimizes the probability of being wrong:

$$f^* = \underset{f:\mathcal{X}\to\{0,1\}}{\text{Argmin}}\ \mathbb{P}(f(X) \neq Y)$$

The classification problem

Suppose our problem is binary and we want to classify the observations into two categories: 0 and 1. In this case the Bayes classifier is the function f^* that minimizes the probability of being wrong:

$$f^* = \underset{f:\mathcal{X}\to\{0,1\}}{\text{Argmin}}\ \mathbb{P}(f(X) \neq Y)$$

Observe that:

$$f^*(x) = \begin{cases} 1 & \text{si} \quad \mathbb{P}(Y=1|X=x) \geq \frac{1}{2} \\ 0 & \text{si} \quad \mathbb{P}(Y=1|X=x) < \frac{1}{2} \end{cases} =$$

The classification problem

Suppose our problem is binary and we want to classify the observations into two categories: 0 and 1. In this case the Bayes classifier is the function f^* that minimizes the probability of being wrong:

$$f^* = \underset{f:\mathcal{X}\to\{0,1\}}{\text{Argmin}}\ \mathbb{P}(f(X) \neq Y)$$

Observe that:

$$f^*(x) = \begin{cases} 1 & \text{si} \quad \mathbb{P}(Y=1|X=x) \geq \frac{1}{2} \\ 0 & \text{si} \quad \mathbb{P}(Y=1|X=x) < \frac{1}{2} \end{cases} = \underset{y\in\{0,1\}}{\text{Argmax}}\mathbb{P}(Y=y|X=x)$$

The classification problem

Another justification of Bayes Classifier:

$$R_L(f) = \underset{f \in \mathcal{C}}{\text{Argmin}} \mathbb{E}\big(L(Y, f(X)\big) = \mathbb{E}_{\mathbf{X}} \left(\sum_{k=1}^{K} L(k, f(x)) \mathbb{P}(Y = k | \mathbf{X} = \mathbf{x}) \right)$$

We have to minimize pointwise this quantity:

$$f(x) = k^* = \underset{k}{\text{Argmin}} \sum_{k=1}^{K} L(k, f(x)) \mathbb{P}(Y = k | \mathbf{X} = \mathbf{x})$$

Suppose we have two classes 1 and 2, then if $k = 1$:

$$L(1,1)\mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x}) + L(2,1)\mathbb{P}(Y = 2|\mathbf{X} = \mathbf{x}) = \mathbb{P}(Y = 2|\mathbf{X} = \mathbf{x}) = 1 - \mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x})$$

and if $k = 2$:

$$L(1,2)\mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x}) + L(2,2)\mathbb{P}(Y = 2|\mathbf{X} = \mathbf{x}) = \mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x}) = 1 - \mathbb{P}(Y = 2|\mathbf{X} = \mathbf{x})$$

then

$$f(x) = \underset{k \in \{1,2\}}{\text{Argmin}} \left(1 - \mathbb{P}(Y = k|\mathbf{X} = \mathbf{x})\right) = \underset{k \in \{1,2\}}{\text{Argmax}} \mathbb{P}(Y = k|\mathbf{X} = \mathbf{x})$$

The classification problem

In the multiclass context, that is when there are more than two categories and the set of labels is $\{1, \ldots, K\}$, the Bayes' classifier is the one that assigns the label k to observation x that has the highest posterior probability, that is to say

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

The classification problem

In the multiclass context, that is when there are more than two categories and the set of labels is $\{1, \ldots, K\}$, the Bayes' classifier is the one that assigns the label k to observation x that has the highest posterior probability, that is to say

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

The Bayes classifier produces the smallest error that any classifier can make. The error if $X = x_0$ is

$$1 - \max_{k \in \{1,\ldots,K\}} \mathbb{P}(Y = k | X = x_0)$$

and therefore the error rate, covering all the possible values x_0 of X is

$$1 - \mathbb{E}\left(\max_{k \in \{1,\ldots,K\}} \mathbb{P}(Y = j | X) \right)$$

The classification problem

In the multiclass context, that is when there are more than two categories and the set of labels is $\{1, \ldots, K\}$, the Bayes' classifier is the one that assigns the label k to observation x that has the highest posterior probability, that is to say

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

The Bayes classifier produces the smallest error that any classifier can make. The error if $X = x_0$ is

$$1 - \max_{k \in \{1,\ldots,K\}} \mathbb{P}(Y = k | X = x_0)$$

and therefore the error rate, covering all the possible values x_0 of X is

$$1 - \mathbb{E}\left(\max_{k \in \{1,\ldots,K\}} \mathbb{P}(Y = j | X) \right)$$

This quantity is called *Bayes error rate*. However, in real problems, not knowing the distribution, it is impossible to calculate Bayes error rate.

Many algorithms try to estimate posterior probabilities and classify the observation in that class that maximize it (KNN, CART, Boosting, etc.)

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

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

The function that minimizes $R_L(f)$ is

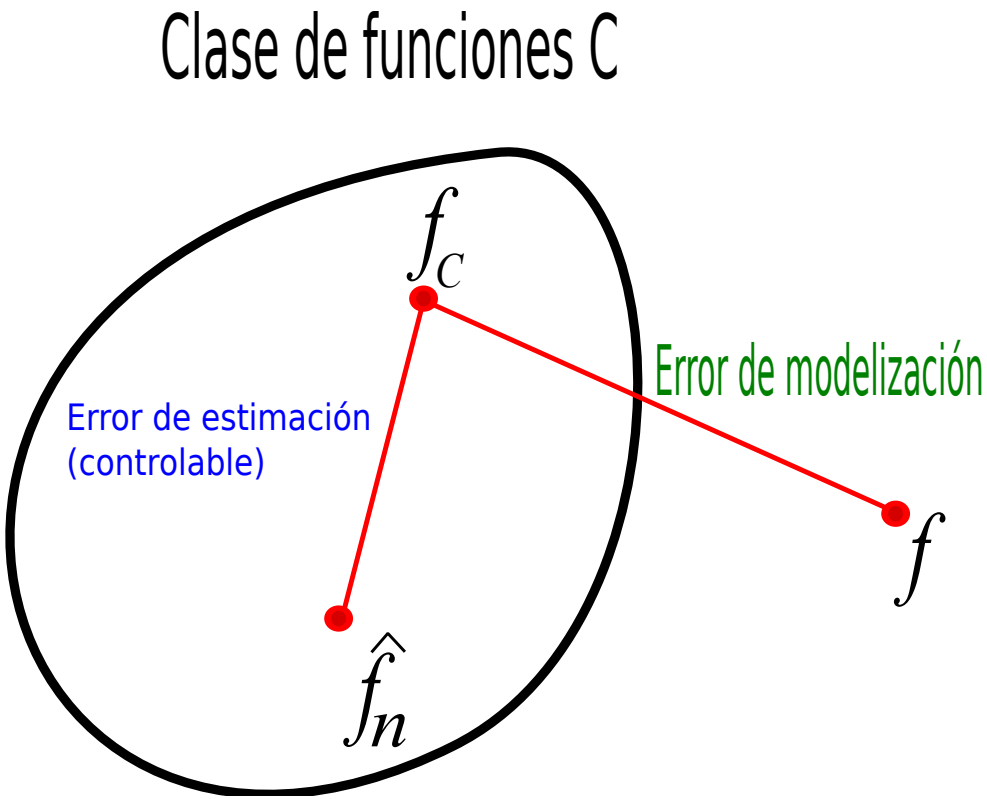
$$f^*(x) = m(x) = \mathbb{E}(Y|X = x)$$

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

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

Plan

1. General Framework and Introduction to Statistical Learning

2. Modelling
   - Choosing the more adequate way
   - Generalization Error
   - Bias-variance trade-off

How estimate f?

The goal is from a sample $\mathcal{L} = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\}$ 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 $\beta_0, \beta_1, \ldots, \beta_p$.

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

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

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.

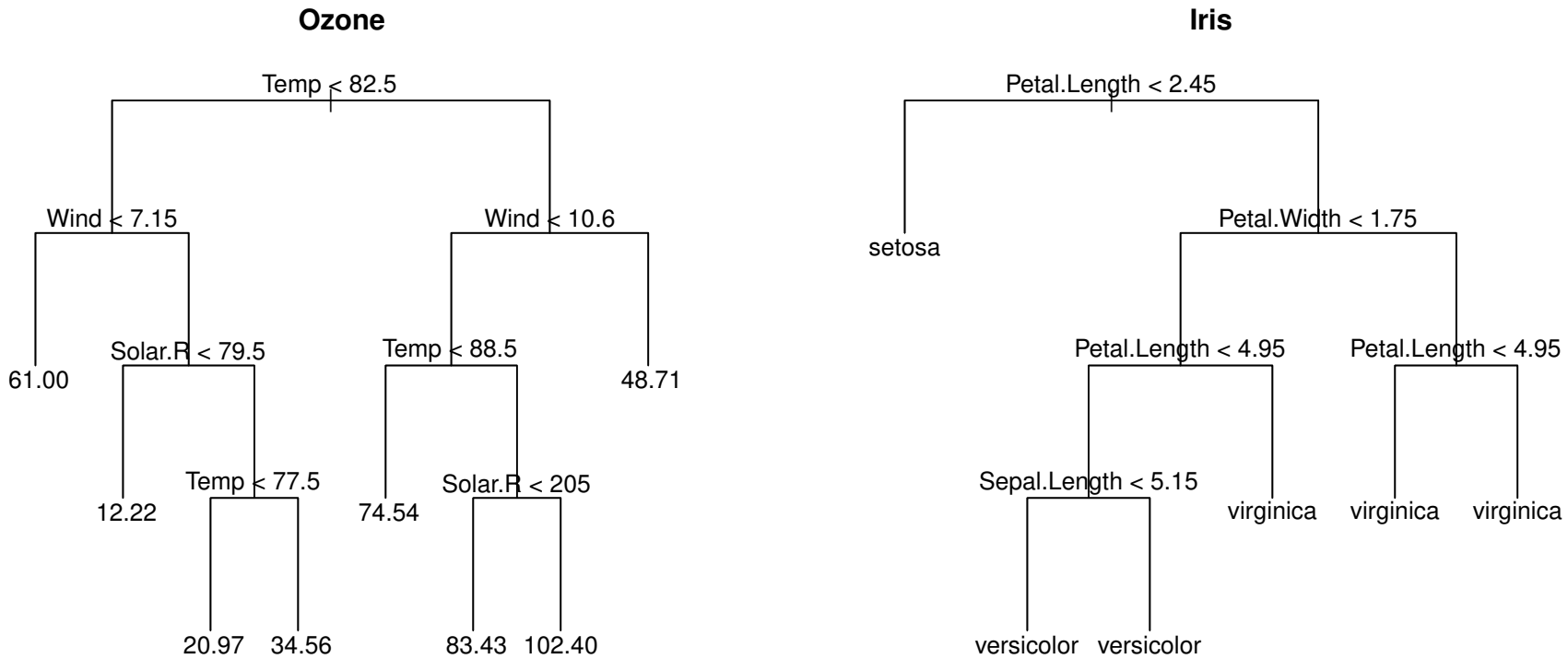
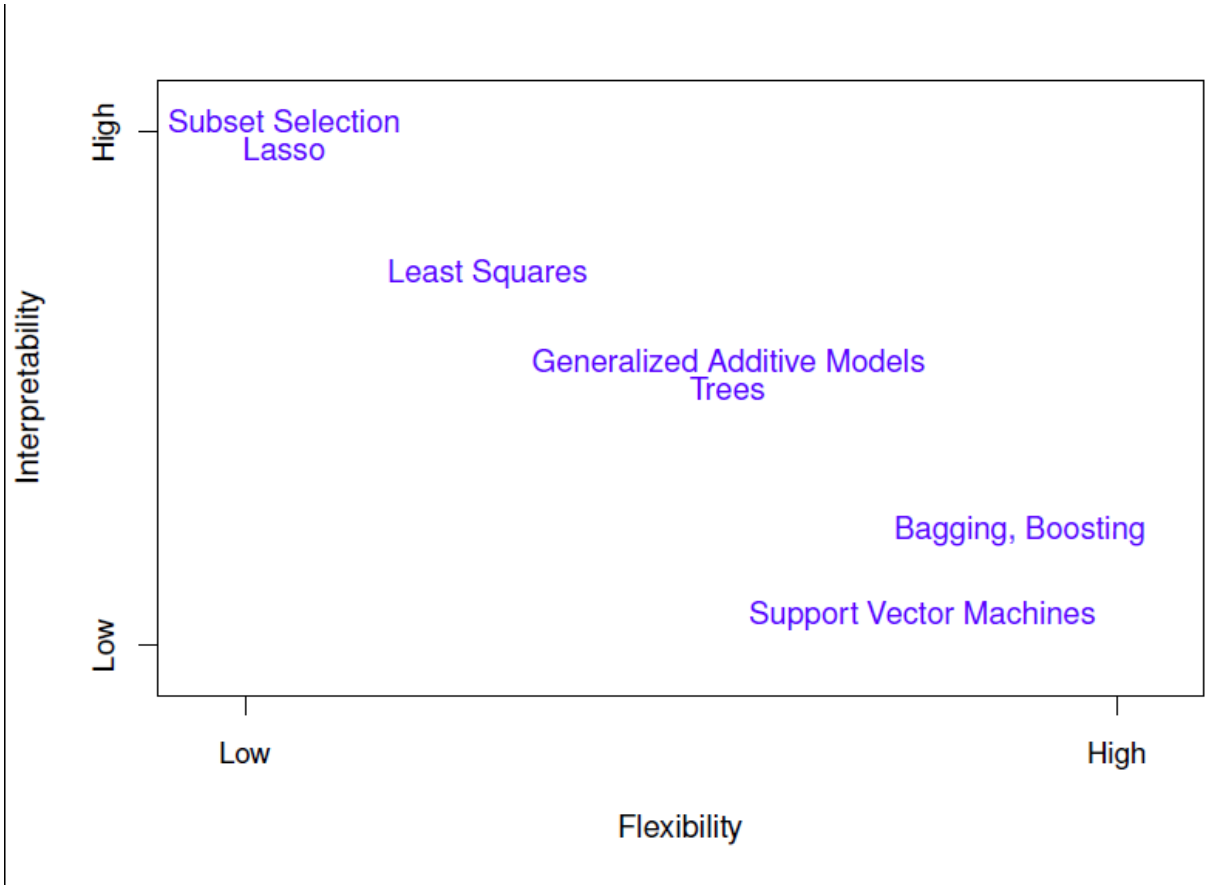


Figure: Classification and Regression Trees (Breiman, 1984)

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* $\mathcal{T} = \{(\mathbf{z_1}, u_1), (\mathbf{z_2}, u_2), \ldots, (\mathbf{z_s}, u_s)\}$ computing the *test*-MSE (generalization error):

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

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* $\mathcal{T} = \{(\mathbf{z_1}, u_1), (\mathbf{z_2}, u_2), \ldots, (\mathbf{z_s}, u_s)\}$ 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

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:

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

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:

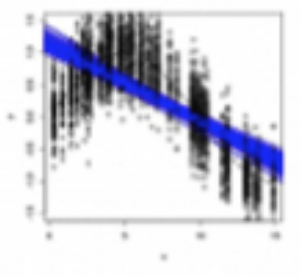
$$\mathbb{E}\left(y_0 - \widehat{f}(x_0)\right)^2 = \text{Var}\left(\widehat{f}(x_0)\right) + \left[\text{Sesgo}\left(\widehat{f}(x_0)\right)\right]^2 + \text{Var}(\epsilon)$$

- As $\text{Var}\left(\widehat{f}(x_0)\right)$ and $\left[\text{Sesgo}\left(\widehat{f}(x_0)\right)\right]^2$ are non negatives, it follows that $\mathbb{E}\left(y_0 - \widehat{f}(x_0)\right)^2$ has as lower bound $\text{Var}(\epsilon)$.
- 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 $X_1, \ldots, X_p$ which clearly has little chance of happening, so the bias will be important. In general, flexible statistical methods have a little bias.

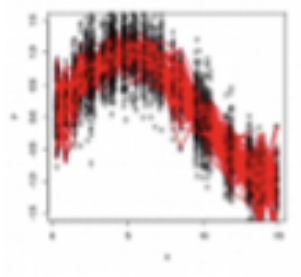
Bias-variance trade-off

Lineal (g1)

Baja Varianza
Gran sesgo



Polinomio g15



Alta Varianza
Bajo sesgo

$$\mathbf{Var}[X] = \mathbf{E}[X^2] - \mathbf{E}[X]^2$$
$$y = f + \epsilon$$

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

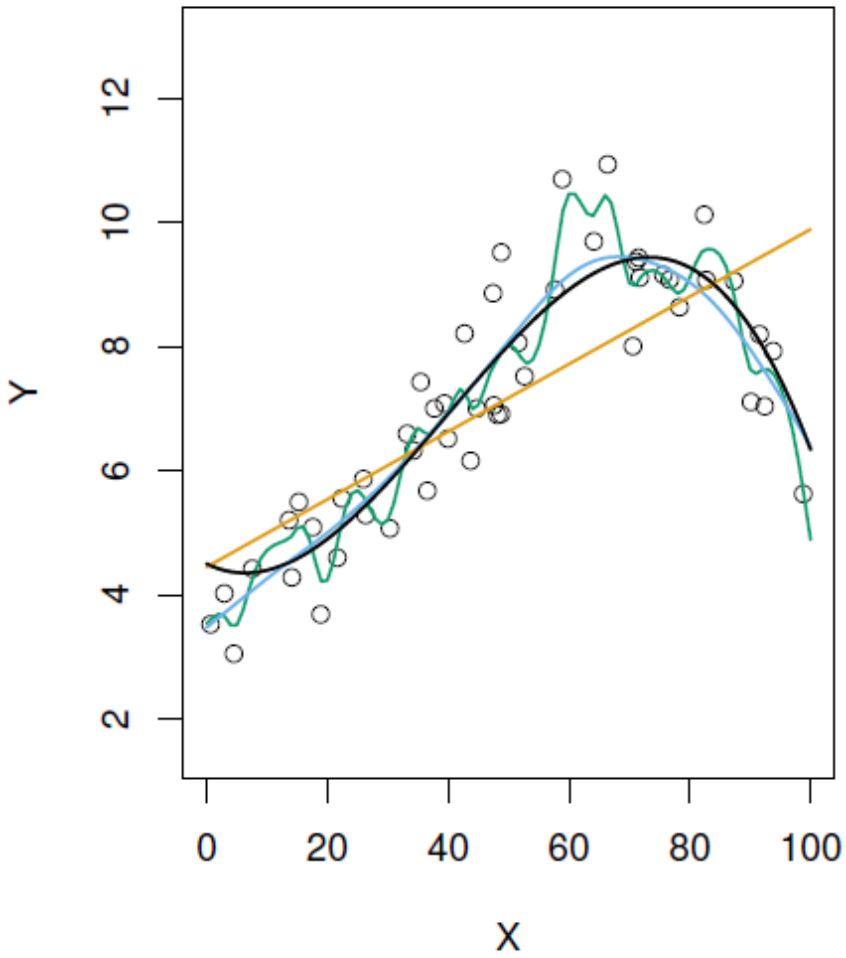
$$\begin{aligned}
\mathbf{E}[(y - \hat{f})^2] &= \mathbf{E}[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= \mathbf{E}[y^2] + \mathbf{E}[\hat{f}^2] - \mathbf{E}[2y\hat{f}] \\
&= \mathbf{Var}[y] + \mathbf{E}[y]^2 + \mathbf{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] \\
&= \mathbf{Var}[y] + \mathbf{Var}[\hat{f}]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

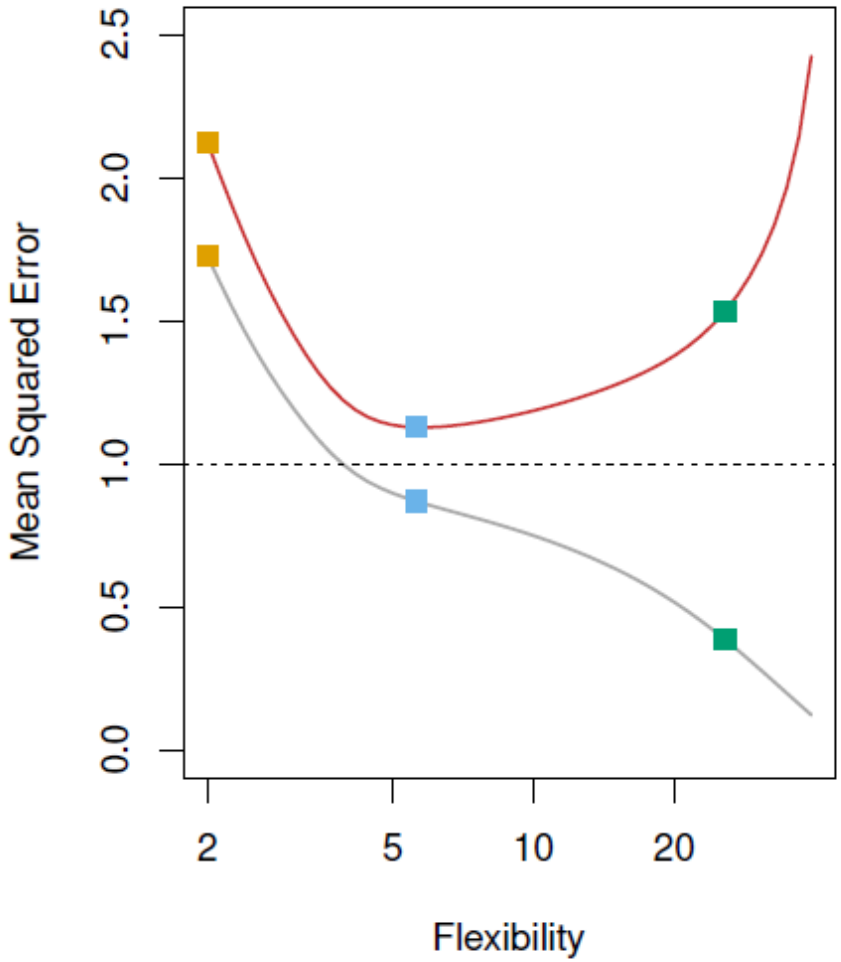
$$= error\ irreducible + varianza(\hat{f}) + Sesgo^2\hat{f}$$

Bias-variance trade-off. Example

Several estimators (smoothing splines) are considered for different data sets (example extracted of James, Witten, Hastie and Tibshirani book).

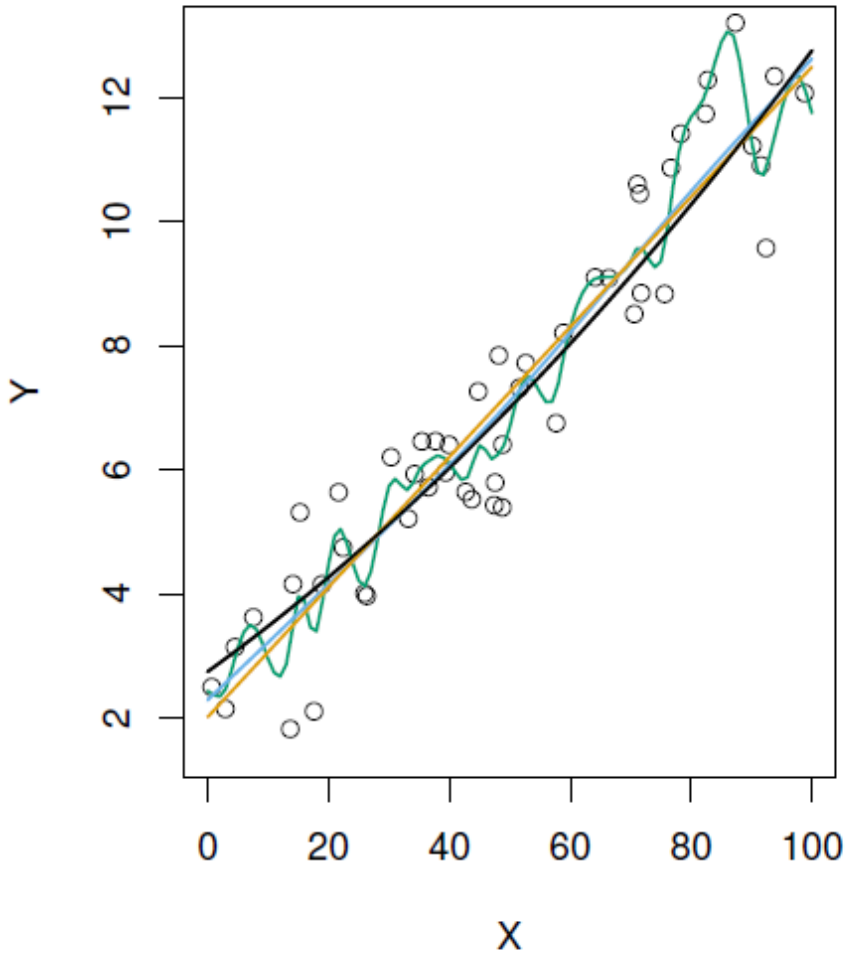
Example 1. On the left hand three estimators with different flexibility adjusting the same data points and on the right hand the MSE curve of the flexibility on the training set (grey) and on a generalization set (red).

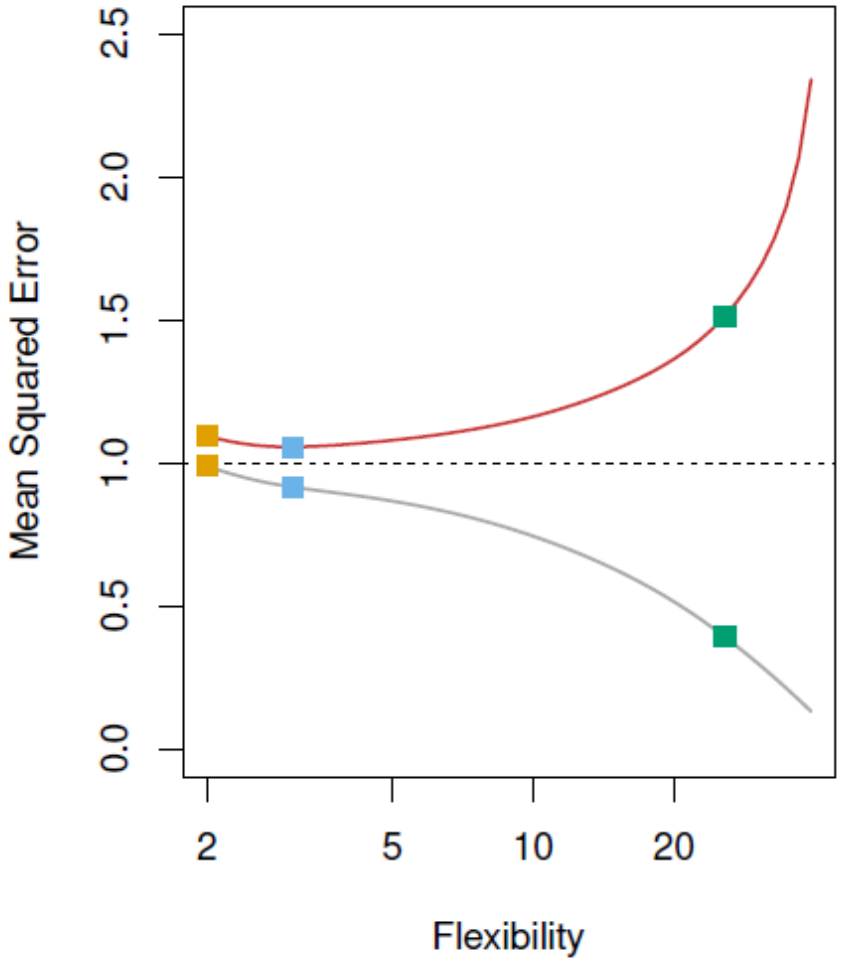




Bias-variance trade-off. Example

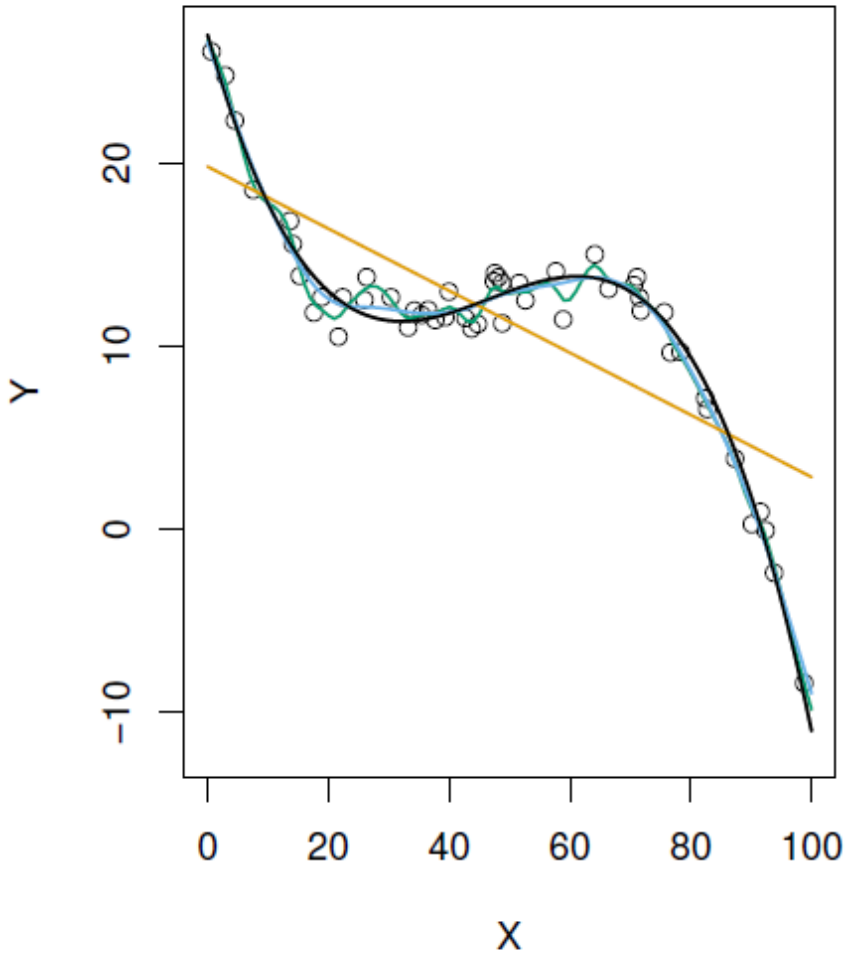
Example 2. On the left hand three estimators with different flexibility adjusting the same data points and on the right hand the MSE curve of the flexibility on the training set (grey) and on a generalization set (red).

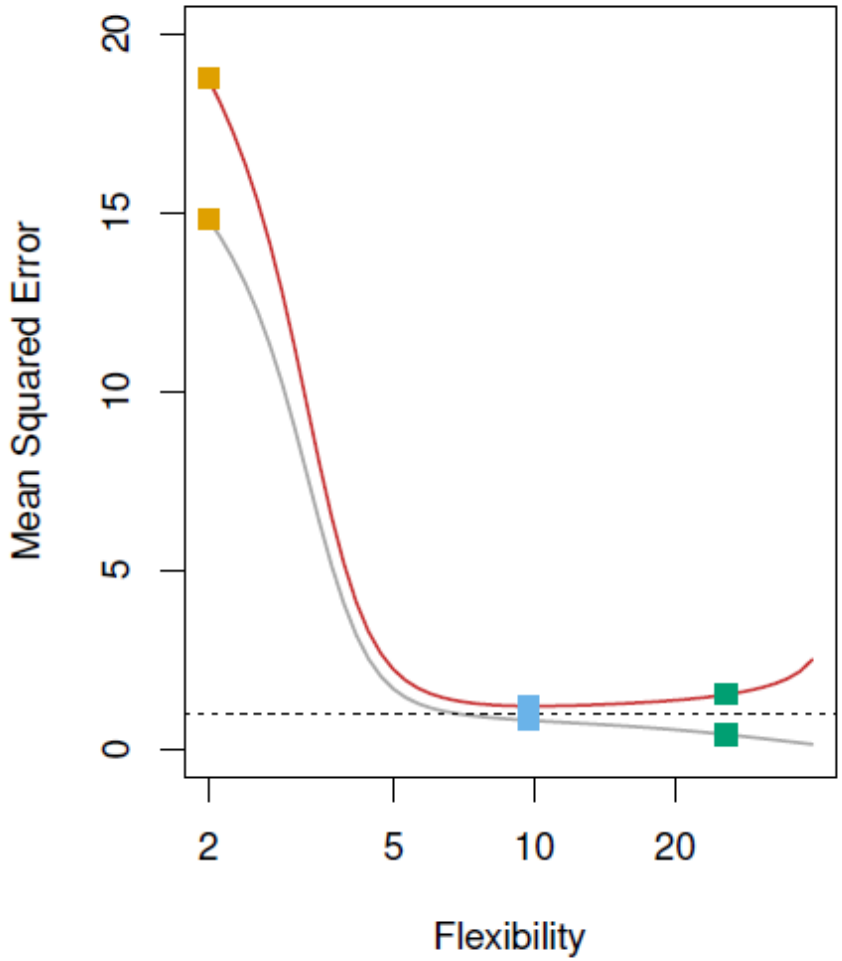




Bias-variance trade-off. Example

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

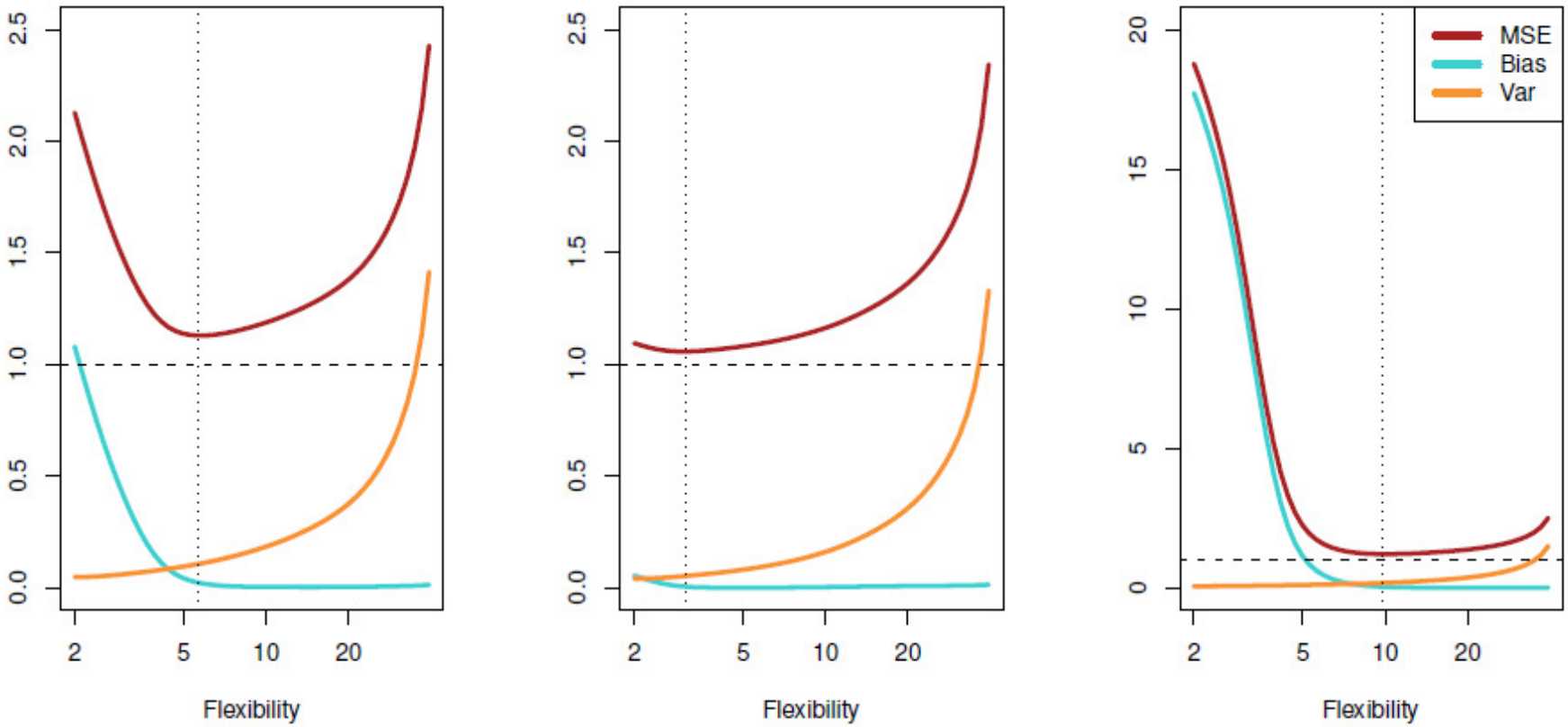


Figure: The three graphs refer to the MSE, bias and variance curves of three previous examples

Bias-variance trade-off. Example

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

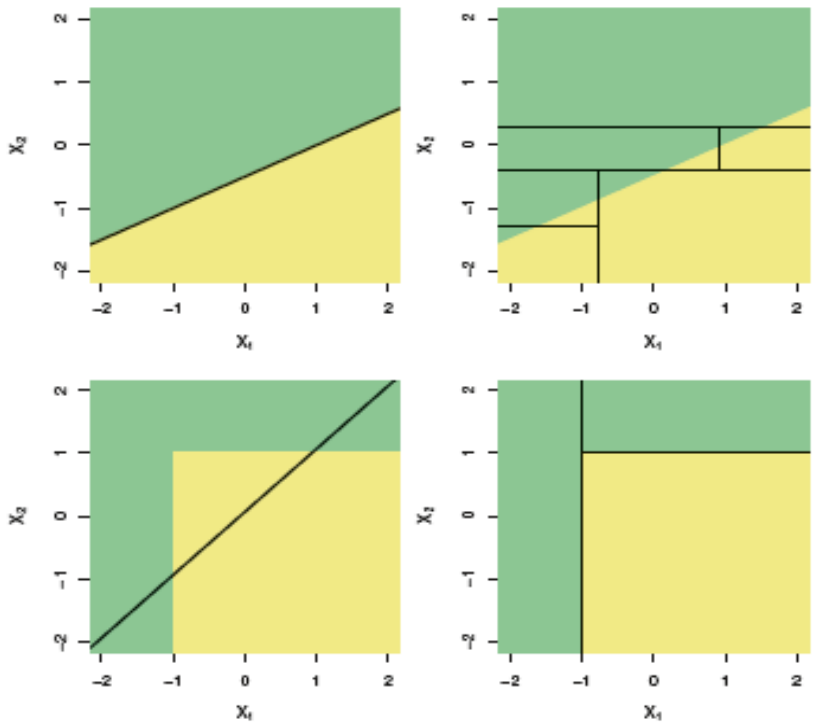


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: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