Business, Economics and Financial Data
Lecture notes

Leonardo Schiavo

Spring 2023

Contents

Disclaimer

These notes are mostly a simple copy-paste of lecture slides or sources. Some topics seen in lecture have been elaborated further and, therefore, there are additions. Often the additions are also a copy-paste from the sources with little reworking. I would caution that this, however, led to no small unevenness in the difficulty of the notes. Also, please be advised that there may be many errors because the goal was only to prepare for my exam and not to write a complete text. Therefore, there will be no future corrected versions of these notes. However, I hope that these notes will be useful to the future reader.

I thank my friends who had the patience to reread to eliminate the serious errors.

Leonardo Schiavo

1 Introduction with additions

1.1 Formal definitions on stochastic processes [2]

Definition 1.1.1. Let T be a subset of $[0, +\inf)$. A family of random variables $\{X_t\}_{t\in T}$, indexed by T, is called a *stochastic process* (or random process). When $T = \mathbb{N}$, $\{X_t\}_{t\in T}$ is said to be a discrete-time process, and when $T = [0, +\inf)$, it is called a continuous-time process.

In this course we treat only stochastic processes that take values in $\mathbb{R}$. Another important definition is the definition of *(strong) stationarity.*

Definition 1.1.2. A stochastic process $\{X_t\}_{t\in\mathbb{N}_0}$ is said to be *stationary* if the random vectors $(X_0, X_1, X_2, ..., X_k)$ and $(X_m, X_{m+1}, X_{m+2}, \ldots, X_{m+k})$ have the same (joint) distribution for all $m, k \in \mathbb{N}_0$.

Definition 1.1.3. Let's define some general functions.

1. *Mean function and variance function*:

$$\mu_t = E[X_t], \quad \sigma_t^2 = Var[X_t], \quad t = 0, 1, 2, \ldots$$

2. *Autocovariance function* $C(t, s) \quad t, s = 0, 1, 2, \ldots$

$$C(t, s) = Cov(X_t, X_s) = E[(X_t - \mu_t)(X_t - \mu_s)]$$

and the function

$$R(t, s) = E[X_t X_s]$$

R and C are symmetric and related through

$$C(t, s) = R(t, s) - \mu_t\mu_s, \quad C(t, t) = \sigma_t^2$$

and, in particular, when $\mu_t \equiv 0$, $C(t, s) = R(t, s)$.

3. *Autocorrelation function*:

$$\rho(t, s) = Corr(X_t, X_s) = \frac{C(t, s)}{\sigma_t\sigma_s} \tag{1}$$

We have that $\rho(t, t) = 1$ and that $|\rho(t, s)| \leq 1$.

4. *Joint probability density* of the vector $(X_{t_1}, \ldots X_{t_n})$

$$f_{t_1,\ldots,t_n}(x_1, \ldots, x_n), \quad t_n \geq t_1$$

5. *Conditional density*

$$f_{t|s}(x|y) = \frac{f_{s,t}(x, y)}{f_s(y)}, \quad t > s$$

6. *Cross-correlation function*: if $(X_n)_{n\geq 0}$ and $(Y_n)_{n\geq 0}$ are two processes we can define

$$C_{X,Y}(t,s) = E[(X_t - E[X_t])(Y_s - E[Y_s])].$$

   measuring of the similarity between two processes, shifted in time.

The functions $C(t,s)$, $R(t,s)$, $\rho(t;s)$ are used to detect repetitions in the process, self-similarities under time shift. For instance, if $(X_n)_{n\geq 0}$ is roughly periodic of period P, $\rho(t+P;t)$ will be significantly higher than the other values of $\rho(t,s)$, (except $\rho(t,t)$ which is always equal to 1). Also a trend is a form of repetitions, self-similarity under time shift, and indeed when there is a trend all values of $\rho(t,s)$ are quite high, compared to the cases without trend.

Definition 1.1.4. A process is called *wide-sense stationary* if μ_t and $R(t+n,t)$ are independent of t.

It follows that in a wide-sense stationary process, also σ_t, $C(t+n,t)$ and $\rho(t+n,t)$ are independent of t. Thus we can speak of mean μ, standard deviation σ, covariance function $C(n) := C(n,0)$, $R(n) := R(n,0)$ and the autocorrelation function $\rho(n) := \rho(n,0)$.

1.2 Time series and empirical quantities [2]

A time series is basically a sequence or real numbers, $x_1,\ldots,x_n$ and also empirical samples have the same form. The name time series is appropriate when the index i of x_i has the meaning of time. A finite realization of a stochastic process is a time series. Ideally, when we have an experimental time series, we think that there is a stochastic process behind. Thus we try to apply the theory of stochastic process. Under the assumptions of stationarity and ergodicity [2] one time series is enough to estimate the true functions of the stochastic process behind the sample. Now consider a time series $x_1,\ldots,x_n$. In the sequel, t and n_t are such $t+n_t=n$. Let us define the following important quantities.

$$\bar{x}_t = \frac{1}{n_t}\sum_{i=1}^{n_t} x_{i+t}, \quad \widehat{\sigma}_t^2 = \frac{1}{n_t}\sum_{i=1}^{n_t}(x_{i+t}-\bar{x}_t)^2$$

$$\widehat{R}(t) = \frac{1}{n_t}\sum_{i=1}^{n_t} x_i x_{i+t}$$

$$\widehat{C}(t) = \frac{1}{n_t}\sum_{i=1}^{n_t}(x_i-\bar{x}_0)(x_{i+t}-\bar{x}_t)$$

$$\widehat{\rho}(t)=\frac{\widehat{C}(t)}{\widehat{\sigma}_0\widehat{\sigma}_t}=\frac{\sum_{i=1}^{n_t}\left(x_i-\bar{x}_0\right)\left(x_{i+t}-\bar{x}_t\right)}{\sqrt{\sum_{i=1}^{n_t}\left(x_i-\bar{x}_0\right)^2\sum_{i=1}^{n_t}\left(x_{i+t}-\bar{x}_t\right)^2}}. \tag{2}$$

These quantities are taken as approximations of

$$\mu_t, \quad \sigma_t^2, \quad R(t,0), \quad C(t,0), \quad \rho(t,0)$$

respectively. In the case of stationary processes, they are approximations of

$$\mu, \quad \sigma^2, \quad R(t), \quad C(t), \quad \rho(t).$$

The empirical correlation function

$$\widehat{\rho}_{X,Y}=\frac{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}}$$

between two sequences $x_1,\ldots,x_n$ and $y_1,\ldots,y_n$ is a measure of their linear similarity.

Remark 1.2.1. We can notice that the formula for the empirical estimation of the autocorrelation function in Equation 2 is circa the same we discussed in the lecture, as the one in the lecture was

$$r_k=\frac{\sum_{t=k+1}^{T}\left(Y_t-\bar{Y}\right)\left(Y_{t-k}-\bar{Y}\right)}{\sum_{t}^{T}\left(Y_t-\bar{Y}\right)^2} \tag{3}$$

We can use the autocorrelation function to study trend and seasonality of a time series. In time series analysis, *trend* refers to the underlying structure in the data that represents a long-term increase or decrease in the series. It can be either linear or non-linear and is independent of any seasonal patterns. On the other hand *seasonality* refers to repeating patterns in data that occur at regular intervals, such as yearly, quarterly, or daily. It represents regular variations in the data that are not due to the trend, but rather due to external factors such as holidays, weather patterns, or economic cycles. Together, trend and seasonality can be used to model and make predictions about future values in a time series.

2 Linear Regression

In this section we will look at three different types of linear regression: simple, multiple and multivariate. The last two are two extensions. Multiple linear regression considers a single dependent variable and many independent variables as regressors. Multivariate linear regression generalizes multiple regression by considering many dependent variables each with many independent variables. Multivariate linear regression is introduced in this section only to give a theoretical basis for generalized additive models.

2.1 Simple linear regression [7]

The *simple linear regression* is a model in which a response variable Y is a linear function of an independent variable x plus random error. Specifically,

$$Y_i = \beta_1 + \beta_2 (x_i - \bar{x}) + \epsilon_i, \quad i = 1, \ldots, n.$$

The independent variables $x_1, \ldots, x_n$ with average $\bar{x}$ are taken to be known constants, β_1 and β_2 are unknown parameters, and $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. from $N(0, \sigma^2)$. This gives a general linear model with design matrix

$$X = \begin{pmatrix} 1 & x_1 - \bar{x} \\ \vdots & \vdots \\ 1 & x_n - \bar{x} \end{pmatrix}$$

In parameterising the mean of Y (called the regression function) as $\beta_1 + \beta_2(x - \bar{x})$, β_1 would be interpreted not as an intercept, but as the value of the regression when $x = \bar{x}$. Note that $\sum_{i=1}^n (x_i - \bar{x}) = \sum_{i=1}^n x_i - n\bar{x} = 0$, which means that the two columns of X are orthogonal. This will simplify many later results. For instance, X will have rank 2 unless all entries in the second column are zero, which can only occur if $x_1 = \cdots = x_n$. Also, since the entries in $X'X$ are inner products of the columns of X, this matrix and $(X'X)^{-1}$ are both diagonal:

$$X'X = \begin{pmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

and

$$(X'X)^{-1} = \begin{pmatrix} 1/n & 0 \\ 0 & 1/\sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

Since

$$X'Y = \begin{pmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n Y_i (x_i - \bar{x}) \end{pmatrix},$$

$$\hat{\beta} = (X'X)^{-1} X'Y = \begin{pmatrix} \frac{1}{n} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n Y_i (x_i - \bar{x}) / \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}.$$

Also,

$$\text{Cov}(\hat{\beta}) = \sigma^2 (X'X)^{-1} = \begin{pmatrix} \sigma^2/n & 0 \\ 0 & \sigma^2/\sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}.$$

To estimate σ^2, since $\hat{\xi}_i = \hat{\beta}_1 + \hat{\beta}_2 (x_i - \bar{x})$,

$$e_i = Y_i - \hat{\beta}_1 - \hat{\beta}_2 (x_i - \bar{x}),$$

280 14 General Linear Model and then

$$S^2 = \frac{1}{n-2}\sum_{i=1}^{n} e_i^2.$$

This equation can be rewritten in various ways. For instance,

$$S^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2\left(1-\hat{\rho}^2\right),$$

where $\hat{\rho}$ is the sample correlation defined as

$$\hat{\rho} = \frac{\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)\left(x_i - \bar{x}\right)}{\left[\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2\right]^{1/2}}.$$

This equation shows that $\hat{\rho}^2$ may be viewed as the proportion of the variation of Y that is "explained" by the linear relation between Y and x. Using (14.22),

$$\left(\hat{\beta}_1 - \frac{St_{\alpha/2,n-2}}{\sqrt{n}}, \hat{\beta}_1 + \frac{St_{\alpha/2,n-2}}{\sqrt{n}}\right)$$

is a $1-\alpha$ confidence interval for β_1, and

$$\left(\hat{\beta}_2 - \frac{St_{\alpha/2,n-2}}{\sqrt{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2}}, \hat{\beta}_2 + \frac{St_{\alpha/2,n-2}}{\sqrt{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2}}\right)$$

is a $1-\alpha$ confidence interval for β_2.

2.2 Multiple linear regression [5]

Definition 2.2.1. The following model is called *multiple linear regression* model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{4}$$

where X_j it the j^{th} predictor and β_j quantifies the relationship between that variable and the response. We interpret β_j as the average effect on Y of a one unit increase in X_j, holding all other predictors fixed.

Given estimates $\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p$ we can make predictions using the formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p. \tag{5}$$

The parameters are estimated through the ordinary least squares method (OLS), by minimizing the deviance

$$S = \sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

over the training set with size n. For other and more in-depth analysis on multiple linear regression and parameter search see [7].

Remark 2.2.2. We make the following assumptions regarding error terms $(\epsilon_1, \dots, \epsilon_n)$

1. Errors have mean zero, i.e. $E[\epsilon_k] = 0$ for every $k = 1, \dots, n$
2. Errors are uncorrelated, i.e. $Cov(\epsilon_i, \epsilon_k) = 0$ for every $i, k = 1, \dots, n$
3. Errors are uncorrelated with $X_{j,k}$, i.e. $Cov(X_{i,j}, \epsilon_k) = 0$ for every $i = 1, \dots, p$ and $j, k = 1, \dots, n$

Note that the index $i = 1, \dots, p$ refers to the single prediction and slices trough all the features, while indices $j, k = 1, \dots, n$ refer to different predictions.

2.2.1 Measuring the goodness of the fit

In order to measure the goodness of fit of the model we introduce the *coefficient of determination* R^2, or *R-square*, which is a statistical measure of the goodness of fit of a regression model, representing the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It ranges from $-\infty$ to 1, where a higher value of R^2 indicates a better fit of the model to the data. "There are cases where R^2 can yield negative values. This can arise when the predictions that are being compared to the corresponding outcomes have not been derived from a model-fitting procedure using those data. Even if a model-fitting procedure has been used, R^2 may still be negative, for example when linear regression is conducted without including an intercept, or when a non-linear function is used to fit the data. In cases where negative values arise, the mean of the data provides a better fit to the outcomes than do the fitted function values, according to this particular criterion." [10] Before giving the formal definition, it is necessary to provide the following terminology.

Definition 2.2.3. Given a data set with n values marked $y = (y_1, ..., y_n)^T$, each y_i associated with a fitted (or modeled, or predicted) value $\hat{y}_i$. We recall that the residuals as are $e_i = y_i - \hat{y}_i$ and that the empirical mean of the observed data is

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Now we can define the quantities

$$\begin{aligned} RSS &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2 \\ TSS &= \sum_{i=1}^{n}(y_i - \bar{y})^2 \\ ESS &= \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 \end{aligned}$$

which are called, respectively, *Residual Sum of Squares*, *Total Sum of Squares* and *Explained Sum of Squares.*

Definition 2.2.4. In the context of the Definition 2.2.3 we can define

$$R^2 = 1 - \frac{RSS}{TSS} \tag{6}$$

Proposition 2.2.5. *In the cases of simple linear regression or ordinary least squares model it holds the following equality:*

$$TSS = ESS + RSS$$

Corollary 2.2.6. *For simple linear regression and ordinary least squares model*

$$R^2 = \frac{ESS}{TSS} \tag{7}$$

2.2.2 Testing the significance of the model and the parameters

In order to test the global significance of the model we perform the following test

$$\begin{aligned} &H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ &H_1 : \text{at least one } \beta_j \neq 0 \end{aligned}$$

through the F statistic

$$F = \frac{ESS/p}{RSS/(n-p-1} = \frac{R^2/p}{(1-R^2)/(n-p-1)}$$

And to evaluate the significance of parameters the hypothesis are

$$\begin{aligned} &H_0 : \beta_j = 0 \\ &H_1 : \beta_j \neq 0 \end{aligned}$$

with the test

$$t = \frac{b_j}{se(b_j)}$$

where b_j is the estimate of the j^{th} coefficient and $se(b_j)$ is the standard error.

2.2.3 Collinearity

Collinearity refers to the situation in which two or more predictor variables are closely related to one another. Effects of collinearity are:

1. reduction the accuracy of estimates of the regression coefficients
2. the standard error for β_j grows
3. the t-statistic declines, thus we may fail to reject $H_0 : \beta_j = 0$

Collinearity can be detected by

- looking at the correlation matrix of the predictors: an element of this matrix that is large in absolute value indicates a pair of highly correlated variables. Note: it is possible for collinearity to exist between three or more variables, this is called multicollinearity;
- the *Variance Inflation Factor* (VIF):

$$VIF_j = \frac{1}{1 - R_j^2} \tag{8}$$

where R_j^2 is the determination index of the regression of the j_{th} variable on the other $k - 1$ predictors. If $R_j^2 = 0$, then $\text{VIF}_j = 1$. If there is a multicollinearity problem, then $\text{VIF}_j > 1$. For example, $R_j^2 = 0.9, \text{VIF}_j = 10$.

2.2.4 Multiple linear regression applied to time series

Many business and economic problems involve the use of time series data. The linear regression model may be usefully employed to model monthly, quarterly or yearly data.

- A linear trend may be easily included through a predictor $X_{1,t} = t$.
- Seasonality may be modeled with seasonal dummy variables. As a general rule, we use $s - 1$ dummy variables to describe s periods (to avoid perfect multicollinearity).

For instance, a model for quarterly data with trend and seasonality may be

$$Y_t = \beta_0 + \beta_1 t + \beta_2 S_2 + \beta_3 S_3 + \beta_4 S_4 + \varepsilon_t \tag{9}$$

Trend and seasonality are modelled as a series of straight lines with different intercept and same slope. The first quarter is described with the model $Y_t = \beta_0 + \beta_1 t$. Parameters $\beta_2, \beta_3, \beta_4$ describe the variation with respect to β_0 due to seasonality.

Note that time series data tend to be autocorrelated, which, in this case, occurs when a the effect of a variable is spread over time. For example, a change in prices may have an effect on both current and future sales. Autocorrelation may be detected through a graphical inspection of residuals or specific tests on residuals.

2.2.5 Autocorrelated residuals

A typical example of autocorrelation is defined as

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t$$

with

$$\epsilon_t = \rho\epsilon_{t-1} + \nu_t$$

where ρ is the correlation between sequential errors and ν_t is an erratic component with mean zero and constant variance. If $\rho = 0$ then $\epsilon_t = \nu_t$. The *Durbin-Watson* test is typically used to diagnose this kind of autocorrelation.

Definition 2.2.7. The system of hypothesis of the *Durbin-Watson* is

$$H_0 : \rho = 0 \quad H_1 : \rho > 0$$

and the *Durbin-Watson test* is defined as

$$DW = \frac{\sum_{t=2}^{n} (e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$

The values of DW range between 0 and 4 with a central value of 2 . For large samples, the following holds

$$DW = 2\,(1 - r_1(e))$$

where $r_1(e)$ is the residual autocorrelation at lag 1. Since $-1 < r_1(e) < 1$, then $0 < DW < 4$. To solve the problem of autocorrelation we need to examine the model: is the functional form correct? Are there any omitted variables?

2.3 Multivariate linear regression [6]

Multivariate linear regression is a natural extension of multiple linear regression in that both techniques try to interpret possible linear relationships between certain input and output variables. Multiple regression is concerned with studying to what extent the behavior of a single output variable Y is influenced by a set of r input variables $\mathbf{X} = (X_1, \cdots, X_r)^\tau$. Multivariate regression has s output variables $\mathbf{Y} = (Y_1, \cdots, Y_s)^\tau$, each of whose behavior may be influenced by exactly the same set of inputs $\mathbf{X} = (X_1, \cdots, X_r)^\tau$.

So, not only are the components of $\mathbf{X}$ correlated with each other, but in multivariate regression, the components of $\mathbf{Y}$ are also correlated with each other (and with the components of $\mathbf{X}$). In this chapter, we are interested in estimating the regression relationship between $\mathbf{Y}$ and $\mathbf{X}$, taking into account the various dependencies between the r-vector $\mathbf{X}$ and the s-vector $\mathbf{Y}$ and the dependencies within $\mathbf{X}$ and within $\mathbf{Y}$.

3 Bass Model and extensions

3.1 (Standard) Bass model [5]

The Bass Model (or Bass Diffusion Model) was developed by Frank Bass. It consists of a simple differential equation that describes the process of how new products get adopted in a population. The model presents a rationale of how current adopters and potential adopters of a new product interact. The basic premise of the model is that adopters can be classified as innovators or as imitators, and the speed and timing of adoption depends on their degree of innovation and the degree of imitation among adopters. The Bass model has been widely used in forecasting, especially new products' sales forecasting and technology forecasting.

Definition 3.1.1. The differential equation that defines the (standard) Bass Model is

$$z'(t) = \left(p + q\frac{z(t)}{m}\right)(m - z(t)) \tag{10}$$

where $z(t)$ is the number of total adopters at time t, $z'(t)$ is the rate of change in the number of adopters at time t, p the coefficient of innovation, q the coefficient of imitation, m the market potential. Hence the quantity qz/m is called "the word of mouth".

3.1.1 Solutions to the (standard) Bass model

First we have to make some reasonable assumptions on the parameters and on the function and we fix $p > 0$, $q \geq 0$, $m > 0$ and the boundary conditions $z(0) = 0$, $z(t) \geq 0$ for all $t \geq 0$, which is the interval of the solution.

We can begin by transforming the Equation 10 with $y = z/m$ obtaining:

$$y'(t) = (p + qy(t))(1 - y(t)) \tag{11}$$

we can see that this is classified as an ordinary differential equation of the first order, autonomous since $f(y) = (p + qy)(1 - y)$ does not depend on t. A first analysis on f gives the local existence and uniqueness of the solution to the Cauchy problem since f is differentiable (thus locally lipschitz). The global solution is then unique for extension of the local solution. Proceeding with the calculations we can separate the variables

$$\frac{dy}{(p+qy)(1-y)} = dt$$

and split into two factors the left side:

$$\frac{dy}{f(y)} = \frac{dy}{(p+qy)(1-y)} = \frac{qdy}{(p+q)(p+qy)} + \frac{dy}{(p+q)(1-y)}$$

integrating both sides we have that

$$\int \frac{dy}{f(y)} = \frac{1}{p+q} \ln \left| \frac{p+qy}{1-y} \right| + c = t \tag{12}$$

Now we exclude the solution in which y is constant (this is not satisfying the original ODE) and we divide into two cases: $y < 1$ and $y > 1$ but only the first one can satisfy the boundary condition $y(0) = 0$, thus it remains only the first case that is going to give the solution.

Isolating y in Equation 12 considering $y < 1$:

$$y = \frac{e^{(p+c)(q+p)} - p}{q + e^{(p+c)(q+p)}}$$

Now with the condition $y(0) = 0$ we have that the constant (which sometimes in the calculations absorbs signs or other constants) c must be

$$c = \frac{ln(p)}{p+q}$$

and the final solutions is

$$y(t) = \frac{e^{t(p+q)} - 1}{\frac{q}{p} + e^{t(p+q)}} \tag{13}$$

In order to obtain the usual solution saw in the lecture and in the literature we just need to rearrange the solution (or multiply numerator and denominator by $e^{-t(p+q)}$):

$$y(t) = F(t;p,q) = \frac{1 - e^{-t(p+q)}}{1 + \frac{q}{p}e^{-t(p+q)}} \tag{14}$$

We can notice that the first derivative of the solution is always positive, hence the solution is strictly increasing up to its limit for $t \to +\inf$ which is 1. It was expected since z can not exceed the market potential.

Returning to the variable z:

$$z(t) = mF(t;p,q) = m\frac{1 - e^{-t(p+q)}}{1 + \frac{q}{p}e^{-t(p+q)}} \tag{15}$$

3.1.2 Properties and problems of the Bass model

The Bass Model is indeed a parsimonious model since it just works with three parameters, m, p and q, and for this reason it is very simple to interpret and it needs only aggregate sales data. On the other side there are some limitations: m is constant during the whole life cycle, we are not accounting for marketing mix strategies and we are doing the hypothesis that the products have a limited life cycle. Furthermore The Bass Model does not account for the effect of exogenous variables, such as marketing mix, public incentives, environmental shocks. Besides, in some cases the diffusion process does not have a bell shape curve, but a more complex structure. For those reasons the *Generalized Bass Model* was introduced.

3.2 Generalized Bass model [5], [4]

Definition 3.2.1. The *Generalized Bass Model* (Bass et al., 1994) adds an intervention function $x(t)$ and the differential equation representing the model is

$$z'(t) = \left(p + q\frac{z(t)}{m}\right)(m - z(t))x(t). \tag{16}$$

where $x(t)$ is an integrable (over $t \geq 0$) and non negative function.

The Bass Model is a special case where $x(t) = 1$ and if $0 < x(t) < 1$ the process slows down, while if $x(t) > 1$ the process accelerates.

We will not calculate the solution explicitly because to find it, it is sufficient to retrace the solution for the standard case by isolating $x(t)$ to the right and integrating. The solution, always for $t \geq 0$, is

$$z(t) = m\frac{1 - e^{-(p+q)\int_0^t x(\tau)d\tau}}{1 + \frac{q}{p}e^{-(p+q)\int_0^t x(\tau)d\tau}} \tag{17}$$

Interestingly, $x(t)$ does not change the market potential m but only the speed of the process. This is not obvious since $x(t)$ represents intervention strategies (or market shocks) and, in theory, could affect the market potential. This actually would also be intended, since those who intervene with advertising or market strategies would wish to increase the number of total sales, however, this is not the case and we can see that in the GBM $x(t)$ does not increase market potential.

Remark 3.2.2. The shape of $x(t)$ can vary with respect to what the user wants to estimate in the time series. Some remarkable options are

1. the *original form* of function $x(t)$ designed by Bass et al. (1994) jointly considers the percentage variation of prices and advertising efforts has the form

$$x(t) = 1 + \Pr(t) + A(t) \tag{18}$$

where $\mathrm{Pr}(t)$ and $A(t)$ are price and advertising at time t.

2. the *exponential shock*

$$x(t) = 1 + c_1 e^{b_1(t-a_1)} I_{t \geq a_1} \tag{19}$$

where parameter c_1 is intensity and sign of the shock, b_1 is the 'memory' of the effect and is typically negative, and a_1 is the starting time of the shock. The use of exponential shock is suitable for identifying the positive effect of marketing strategies or incentive measures, in order to speed up the diffusion process. Also, a negative shock may represent a fast slowdown in sales due to the entrance of a competitor.

3. A more stable shock, acting on a longer period of time, may be modeled through a *rectangular shock*

$$x(t) = 1 + c_1 I_{t \geq a_1} I_{t \leq b_1}, \tag{20}$$

where parameter c_1 describes intensity of the shock, either positive or negative, parameters a_1 and b_1 define beginning and end of the shock (con $a_1 < b_1$). The rectangular shock is useful to identify the effect of policies and measures within a limited time interval.

4. A *mixed shock* since it may be useful to have more than one shock of different nature. A simple case is made of a couple of shocks, rectangular and exponential,

$$x(t) = 1 + c_1 I_{t \geq a_1} I_{t \leq b_1} + c_2 e^{b_2(t-a_2)} I_{t \geq a_2} \tag{21}$$

3.3 Dynamic market potential [5], [4]

In the context of bass model extensions if we try to use a non-constant market potential the idea of Guseo and Guidolin in [8] is the differential equation:

$$z'(t) = m(t) \left\{ \left(p + q\frac{z(t)}{m} \right) \left(1 - \frac{z(t)}{m(t)} \right) \right\} + z(t)\frac{m'(t)}{m(t)} \tag{22}$$

In this context, however, we immediately notice that the model is reduced to a standard bass model, in fact

$$\frac{z'(t)m(t) - z(t)m'(t)}{m^2(t)} = \left(\frac{z(t)}{m(t)} \right)' = \left(p + q\frac{z(t)}{m(t)} \right) \left(1 - \frac{z(t)}{m(t)} \right)$$

and, by setting $y(t) = z(t)/m(t)$, we have

$$y'(t) = p + qy(t)(1 - y(t))$$

which is a standard Bass Model. And, thus, the solution to the Dynamic Market Potential version of the Bass Model is

$$z(t) = m(t)F(t) = m(t)\frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}} \tag{23}$$

Remark 3.3.1. We can observe that market of new products is unstable and uncertain in the first phase of diffusion, which is called "incubation". Advertising and promotional efforts play a central role to overcome this phase. These efforts influence the structure of the market potential, which depends on information on the product. Communication and adoption are two separate phases, needing a distinct modelling. In order to solve this problem the Guseo Guidolin Model was introduced in [8].

3.3.1 Guseo Guidolin model

In Guseo and Guidolin (2009) [8] $m(t)$ is made depend on a communication process about the new product, which typically precedes the adoption phase and serves the purpose of building the market, so that the final model, called GGM, takes the form

$$z(t) = K\sqrt{\frac{1 - e^{-(p_c+q_c)t}}{1 + \frac{q_c}{p_c}e^{-(p_c+q_c)t}}}\frac{1 - e^{-(p_s+q_s)t}}{1 + \frac{q_s}{p_s}e^{-(p_s+q_s)t}}. \tag{24}$$

In the GGM cumulative adoptions $z(t)$ depends on parameters K, p_c, q_c, p_s, q_s. Parameters p_c and q_c describe the dynamics of spread of information, contributing to the creation of the market potential, while parameters p_s and q_s refer to the adoption phase. The BM is a special case of GGM where the spread of information is so fast that there is a set of potential adopters ready to purchase as soon as the product enters the market, $m(t) = K$

3.4 Multivariate diffusion models: UCRCD Model [4]

By generalizing to the multivariate case the basic structure of the BM, diffusion models under duopolistic competition have been developed. A common feature of these models is accounting for the interplay between products by splitting the imitation effect into two parts: the within-product imitation, due to a product's specific sales, and the cross-product imitation, due to sales of the concurrent. In addition, competitors may enter the market at the same time so that their life-cycles are essentially simultaneous, or, more generally, a product starts as a monopolist and gains concurrent brands along the way. The situation of sequential market entry, also called diachronic competition, is more common in reality, although it is less treated in literature. The UCRCD model by Guseo and Mortarino (2014) postulates a diffusion process characterized by two phases: monopoly and competition. Borrowing some terms from game theory, the first market player, may be termed the incumbent, while the second, entering the market at a second stage, may be referred as the entrant. Given these different phases, the market potential may have different levels: m_a, the market potential of the incumbent in the monopolistic phase, and m_c, the market potential under competition. The residual market $m - z(t)$ is assumed to be common, where $z(t) = z_1(t) + z_2(t)$ are common cumulative adoptions. The second market player enters the market at time $t = c_2$ with $c_2 > 0$. The model is a system of differential equations where $z_1'(t)$ and $z_2'(t)$ indicate instantaneous adoptions of the first and of the second market player, respectively, and I_A is an indicator function of event A,

Definition 3.4.1. This framework defines the *Unbalanced competition and regime change diachronic model* with the system of differential equations:

$$\begin{aligned} z_1'(t) =& m\left\{\left[p_{1a} + q_{1a}\frac{z(t)}{m}\right](1 - I_{t>c_2}) \right. \\ &\left. + \left[p_{1c} + (q_{1c} + \delta)\frac{z_1(t)}{m} + q_{1c}\frac{z_2(t)}{m}\right] I_{t>c_2}\right\}\left[1 - \frac{z(t)}{m}\right], \end{aligned}$$

$$\begin{aligned} z_2'(t) &= m\left[p_2 + (q_2 - \gamma)\frac{z_1(t)}{m} + q_2\frac{z_2(t)}{m}\right]\left[1 - \frac{z(t)}{m}\right] I_{t>c_2}, \\ m &= m_a(1 - I_{t>c_2}) + m_c I_{t>c_2} \\ z(t) &= z_1(t) + z_2(t) I_{t>c_2}. \end{aligned}$$

q_{1c}	$q_2 - \gamma$	interpretation
negative	negative	full competition
negative	positive	2 competes with 1, 1 collaborates with 2
positive	negative	2 collaborates with 1, 1 competes with 2
positive	positive	full collaboration

Table 1: Table: Sign of cross-imitation coefficients: competition-collaboration

In the monopolistic phase, $t \leq c_2$, the trajectory of the incumbent, $z_1'(t)$, is described according to a standard Bass model with parameters p_{1a}, q_{1a}, and m_a. When $t > c_2$, both concurrents exist in the market and influence each other. The incumbent is characterized by new parameters: the innovation coefficient under competition, p_{1c}, and the imitation coefficient, which is divided into two parts, the within imitation coefficient $q_{1c} + \delta$, measuring internal growth through the ratio z_1/m, and the cross imitation one, q_{1c} which is powered by z_2/m and measures the effect of the diffusion of the entrant on the incumbent. The entrant has three corresponding parameters: the innovation coefficient p_2, the within imitation coefficient q_2, modulating internal growth through the ratio z_2/m and the cross imitation coefficient $q_2 - \gamma$, which measures the effect, of the incumbent. Typically parameters δ and γ are assumed to be different, and the model is called unrestricted UCRCD. If the restriction $\delta = \gamma$ applies, the model takes a reduced form, called standard UCRCD, see Guseo and Mortarino (2014), and a symmetric behavior between the two competitors is assumed. A possible generalization of the UCRCD model has been proposed in Guidolin and Guseo (2015) and Guidolin and Guseo (2020), with a Lotka-Volterra model with churn effects.

3.5 Statistical inference for diffusion models [4]

The statistical implementation of diffusion models is quite sensitive to the amount of data available and reliable estimates are obtained if non-cumulative data include the peak, as observed by Srinivasan and Mason (1986). However, this clearly reduces model usefulness for forecasting. Mahajan et al. (1990) effectively synthesized the problem, stating that 'parameter estimation for diffusion models is primarily of historical interest; by the time sufficient observations have been developed for reliable estimation, it is too late to use the estimates for forecasting purposes'. Van den Bulte and Lilien (1997) considered some bias in parameter estimation, including the tendency to underestimate the market potential, whose value is generally close to the latest observed data. Estimation aspects were also discussed in Venkatesan and Kumar (2002), Venkatesan et al. (2004) and Jiang et al. (2006). Empirical experience demonstrated that ordinary least squares technique (OLS) is non-optimal for estimating diffusion models, because of some shortcomings including the tendency to yield negative sign parameters. Srinivasan and Mason (1986) proposed using the nonlinear least squares approach (NLS), which is generally accepted to be the more reliable method of estimation. Specifically, the structure of a nonlinear regression model, following Seber and Wild (1989), may be considered

$$w(t) = \eta(\beta, t) + \varepsilon(t), \tag{25}$$

where $w(t)$ is the observed response, $\eta(\beta, t)$ is the deterministic component describing instantaneous or cumulative processes, depending on parameter set β and time t, and $\varepsilon(t)$ is a residual term, not necessarily independent and identically distributed (i.i.d.).

Model global goodness-of-fit is evaluated through the R^2, the value of which is typically greater than 0.95, because it is calculated on cumulative data. From this perspective,

comparison and selection between concurrent models becomes essential. The performance of an extended model, m_2, compared with a nested one, m_1, may be evaluated through a squared multiple partial correlation coefficient $\tilde{R}^2$ in the interval $[0, 1]$, namely,

$$\tilde{R}^2 = \frac{R^2_{m_2} - R^2_{m_1}}{1 - R^2_{m_1}} \tag{26}$$

where $R^2_{m_i}, i = 1, 2$ is the standard determination index of model m_i. The $\tilde{R}^2$ coefficient has a monotone correspondence with the F-ratio

$$F = \frac{\tilde{R}^2(n - v)}{\left(1 - \tilde{R}^2\right) u} \tag{27}$$

where n is the number of observations, v the number of parameters of the extended model m_2, and u the incremental number of parameters from m_1 to m_2. Under strong conditions on the distributional shape of the error term $\varepsilon(t)$, particularly i.i.d. and normality, the statistic F-ratio, for the null hypothesis of equivalence of the two models, is a central Snedecor's F with u degrees of freedom for the numerator and $n - v$ degrees of freedom for the denominator, $F \sim F_{u,n-v}$.

Remark 3.5.1. If R^2 is greater than 0.95 then you can compare two nested models using the $\tilde{R}^2$. If the R^2 is negative it means that model is worse than the baseline model, so there is no point in using that model, let alone a nested model of it, let alone comparing them.

4 Time series models

Let us define a forecasting error $e_t = Y_t - F_t$. We may then define some forecasting accuracy measures: Mean Error, Mean Absolute Error, Mean Squared Error

$$\begin{aligned}
\text{ME} &= \frac{1}{n}\sum_{t=1}^{n} e_t \\
\text{MAE} &= \frac{1}{n}\sum_{t=1}^{n} |e_t| \\
\text{MSE} &= \frac{1}{n}\sum_{t=1}^{n} e_t^2.
\end{aligned}$$

The value of ME, MAE, MSE depend on the scale of data. This makes difficult to compare different models. We may define the percentage error and related measures.

$$\begin{aligned}\mathrm{PE}_t &= \frac{Y_t - F_t}{Y_t}100\\ \mathrm{MPE} &= \frac{1}{n}\sum_{t=1}^{n}\mathrm{PE}_t\\ \mathrm{MAPE} &= \frac{1}{n}\sum_{t=1}^{n}|\mathrm{PE}_t|\end{aligned}$$

4.1 Short-term forecasting [5], [3]

4.1.1 Simple exponential smoothing

In order to perform a short-term forecasting we can implement a *simple exponential smoothing.*

Definition 4.1.1. Given a time series $\{Y_t\}_{t\geq 0}$ we define the *simple exponential smoothing* as

$$F_{t+1} = F_t + \alpha\left(Y_t - F_t\right) = \alpha Y_t + (1-\alpha)F_t \tag{28}$$

where α is a constant term taking values between 0 e 1 included.

The new forecast F_{t+1} is the old forecast F_t with an adjustment. We can see the new forecast F_{t+1} as a weighted average (convex combination) of the last observation, Y_t, and the last forecast, F_t. Observe that the initialization of the process $F_2 = \alpha Y_t + (1-\alpha)F_1$ is problematic since F_1 is not available, typically we use the first observation, $Y_1 = F_1$.

A crucial point in exponential smoothing concerns choosing a suitable value for α, in fact a higher value for α is more sensitive to a change in the data structure, while a lower value generates a 'flat' forecast. Usually a suitable selection for α is that minimizing the MSE.

Remark 4.1.2. If we carry out a few iterations it is clear the meaning of the name "exponential smoothing":

$$\begin{aligned}F_{t+1} &= \alpha Y_t + (1-\alpha)\left[\alpha Y_{t-1} + (1-\alpha)F_{t-1}\right]\\ &= \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + (1-\alpha)^2 F_{t-1}\end{aligned}$$

so that we obtain

$$\begin{aligned}F_{t+1} =& \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2}\\ &+ \alpha(1-\alpha)^3 Y_{t-3} + \ldots + \alpha(1-\alpha)^{t-1}Y_1 + (1-\alpha)^t F_1\end{aligned}$$

4.1.2 Holt's linear trend method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

$$\begin{aligned} \text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\ \text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right) \\ \text{Trend equation} \quad & b_t = \beta^* \left(\ell_t - \ell_{t-1}\right) + \left(1 - \beta^*\right) b_{t-1} \end{aligned}$$

where ℓ_t denotes an estimate of the level of the series at time t, b_t denotes an estimate of the trend (slope) of the series at time t, α is the smoothing parameter for the level, $0 \leq \alpha \leq 1$, and β^* is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$. (We denote this as β^* instead of β for reasons that will be explained in section 7.5.) As with simple exponential smoothing, the level equation here shows that ℓ_t is a weighted average of observation y_t and the one-step-ahead training forecast for time t, here given by $\ell_{t-1}+b_{t-1}$. The trend equation shows that b_t is a weighted average of the estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_{t-1}, the previous estimate of the trend. The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

Remark 4.1.3. The equation we was in the lecture was identical but with different notation:

$$\begin{aligned} L_t &= \alpha Y_t + (1-\alpha)\left(L_{t-1} + b_{t-1}\right) \\ b_t &= \beta\left(L_t - L_{t-1}\right) + (1-\beta) b_{t-1} \\ F_{t+m} &= L_t + b_t m \end{aligned}$$

L_t denotes an estimate of the level of the series at time t and b_t an estimate of the slope t. We can notice that

1. this exponential smoothing is a double smoothing;
2. the forecast function is no longer flat but trending;
3. the m-step-ahead forecast is equal to the last estimated level plus times the last estimated trend value, hence the forecasts are a linear function of m.

4.1.3 Damped trend methods

The forecasts generated by Holt's linear method display a constant trend (increasing or decreasing) indefinitely into the future. Empirical evidence indicates that these methods tend to over-forecast, especially for longer forecast horizons. Motivated by this observation, Gardner & McKenzie (1985) introduced a parameter that "dampens" the trend to a flat line some time in the future. Methods that include a damped trend have proven to be very successful, and are arguably the most popular individual methods when forecasts are required automatically for many series. In conjunction with the smoothing parameters α and β^* (with values between 0 and 1 as in Holt's method), this method also includes a damping parameter $0 < \phi < 1$

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + \left(\phi + \phi^2 + \cdots + \phi^h\right) b_t \\
\ell_t &= \alpha y_t + (1-\alpha)\left(\ell_{t-1} + \phi b_{t-1}\right) \\
b_t &= \beta^* \left(\ell_t - \ell_{t-1}\right) + (1-\beta^*)\, \phi b_{t-1}
\end{aligned}$$

If $\phi = 1$, the method is identical to Holt's linear method. For values between 0 and $1, \phi$ dampens the trend so that it approaches a constant some time in the future. In fact, the forecasts converge to $\ell_T + \phi b_T/(1-\phi)$ as $h \to \infty$ for any value $0 < \phi < 1$. This means that short-run forecasts are trended while long-run forecasts are constant.

In practice, ϕ is rarely less than 0.8 as the damping has a very strong effect for smaller values. Values of ϕ close to 1 will mean that a damped model is not able to be distinguished from a non-damped model. For these reasons, we usually restrict ϕ to a minimum of 0.8 and a maximum of 0.98.

4.2 ARIMA models [5]

ARIMA models provide a typical approach to time series forecasting. Exponential smoothing and ARIMA models are the two most widely used approaches to time series forecasting, and provide complementary approaches to the problem. While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data.

4.2.1 Differenciating

A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary. *Differencing* can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

Definition 4.2.1. We define *Differenciating* by the following equation:

$$y'_t = y_t - y_{t-1}. \tag{29}$$

Observe that usually Seasonal differencing (for monthly data) is

$$y'_t = y_t - y_{t-12}.$$

And further differencings may be performed

$$y^*_t = y'_t - y'_{t-1} = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) .$$

In this framework it is useful to introduce the *Backward shift operator.*

Definition 4.2.2. We define the *Backward shift operator B* as

$$By_t = y_{t-1} \tag{30}$$

In other words, B has the effect of shifting the data back one period.

Observe that two applications of B shifts the data back two periods

$$B\left(By_t\right) = B^2 y_t = y_{t-2}$$

Remark 4.2.3. The backward shift operator is convenient for describing the process of differencing. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1-B)y_t$$

Similarly, if second-order differences have to be computed, then

$$\begin{aligned} y''_t &= \left(y'_t - y'_{t-1}\right) \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \\ &= \left(1 - 2B + B^2\right) y_t \\ &= (1-B)^2 y_t \end{aligned}$$

4.2.2 Autoregressive models

Autoregressive Models are just multiple regression models that are using past values as predictors.

Definition 4.2.4. Recalling the Equation 9 we have the equation for an $AR(p)$ model:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \ldots + \beta_p y_{t-p} + \varepsilon_t \tag{31}$$

which is an *Autoregressive* model of order p.

This is like a multiple regression but with lagged values of y_t as predictors.

4.2.3 Moving average models

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

Definition 4.2.5. We can define the *Moving Average Model* of order q, $MA(q)$, by the following equation:

$$y_t = \beta_0 + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \ldots + \beta_q \varepsilon_{t-q} + \varepsilon_t \tag{32}$$

4.2.4 Autoregressive Integrated Moving Average Models

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for *AutoRegressive Integrated Moving Average*, ARIMA (p, d, q) where p refers to the AR part, q refers to the MA part and d is the degree of first differencing involved.

Example 4.2.6. Notice that a White Noise model $y_t = c + \varepsilon_t$ is an ARIMA$(0, 0, 0)$, while a Random Walk $y_t = y_{t-1} + \varepsilon_t$, is an ARIMA $(0, 1, 0)$

An ARMA(p, q) may be expressed as

$$y_t = c + \phi_1 y_{t-1} + \ldots + \phi_q y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{33}$$

or, by using backshift notation,

$$(1 - \phi_1 B - \ldots - \phi_p B^p)\, y_t = c + (1 + \theta_1 B + \ldots + \theta_q B^q)\, \varepsilon_t \tag{34}$$

Remark 4.2.7. If an ARMA(p, q) model is non stationary, we obtain an ARIMA (p, d, q) model.

Example 4.2.8. The simplest case, ARIMA $(1, 1, 1)$, is defined as

$$(1 - \phi_1 B)\,(1 - B) y_t = c + (1 + \theta_1 B)\, \varepsilon_t$$

4.2.5 ARIMA and seasonality

A further extension to ARMA models concerns seasonality. An ARIMA model with seasonal components is an ARIMA $(p, d, q)(P, D, Q)_s$, where (p, d, q) indicates the non-seasonal part of the model, while (P, D, Q) indicates the seasonal part of order s.

Example 4.2.9. The ARIMA model $(1, 1, 1)(1, 1, 1)_4$ is

$$(1 - \phi_1 B)\left(1 - \Phi_1 B^4\right)(1 - B)\left(1 - B^4\right) y_t = (1 + \theta_1 B)\left(1 + \Theta_1 B^4\right) \varepsilon_t$$

5 Non parametric models

5.1 Bias-Variance trade-off [5], [1]

In general given n couples (x_i, y_i), for $i = 1, \dots, n$, of data we wish to obtain a rule (model), like $\hat{y} = \hat{f}(x)$, that enables us to predict y once we know x. Where $f(x)$ is the underlying true generating function of the data.

One can try to fit a polynomial function

$$f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_p x^p$$

to the data, but how to choose the right degree p? The usual procedures include slitting into train and test set, cross-validation and information criteria. For the first and the second method in this context we simply mention the relationship between bias, variance and mean squared error:

$$\mathbb{E}\left\{\left[\hat{y} - f\left(x'\right)\right]^2\right\} = \text{ bias }^2 + Var\{\hat{y}\} \tag{35}$$

where $\text{ bias } = \mathbb{E}\{\hat{y}\} - f\left(x'\right)$

Proof. All that remains is to perform the calculations by adding and subtracting the factor $\mathbb{E}\{\hat{y}\}$ within the square, expanding the square and remembering that in the double product there is a constant value so we can eliminate the whole double product in expectation by obtaining the thesis. □

Recall that during training the MSE decreases strictly monotonically as epochs and model complexity increase. This tendency leads to the possibility of *overfitting*, i.e., perfect prediction on training data, but very poor generalization ability and thus very high variance when trying to predict on new data. It is clear, therefore, that as the MSE decreases, so as the *inductive bias* of the model also decreases, in general, the variance of the model increases: this phenomenon is known as the *bias-variance trade-off*. By *inductive bias* we refer to the amount of information that the model fails to capture because of its inadequacy or simplicity. To make up for this effect and find the right tradeoff, one can divide the sampling set into *training set* and *validation set*, train the model on the training set, and then evaluate the generalization ability of the model on the validation set. Finally, one chooses the model that has the lowest MSE (or other desired metric) on the validation set ensuring, at least in theory, good generalization ability.

Regarding the third the direct approach would be to inspect the deviance of the residuals but it is too optimistic evaluation of the prediction error. Hence we decide to penalize the deviance: let D be the deviance

$$D = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{36}$$

we perform a monotonic transform

$$-2 \log L = n \log(D/n) + (\text{constant}) \tag{37}$$

Definition 5.1.1. We can define a wide variety of information criteria

$$IC(p) = -2 \log L + \text{penalty}(p) \tag{38}$$

where the choice of the specific penalty function identifies a particular criterion.

We can use those information criteria by choosing p that minimizes $IC(p)$.

criterion	author	penalty (p)
AIC	Akaike	$2p$
AlC$_c$	Sugiura, Hurvich-Tsay	$2p + \frac{2p(p+1)}{n-(p+1)}$
BIC/SIC	Akaike, Schwarz	$p \log n$
HQ	Hannan-Quinn	$cp \log \log n, \quad (c > 2)$

Table 2: Some possibile penalty are in the following table

Remark 5.1.2. It is worth noting that:

1. The $logL$, for the gaussian model, has an interpretation as log-likelihood. Because Equation 37 you get it with the log-likelihood of a normal distribution, hence Gaussian
2. The information criteria in Table 2 are applied also to not nested and not gaussian models.

5.2 K-Nearest Neighbors regression [5]

A first note on the difference between *classification* and *regression*. We speak of classification when in a problem we need to predict to which class, among a finite number of classes, a new object belongs; whereas we speak of regression when there is an underlying continuous function f(x) that generated the data and we want to estimate the value of f on a new input x. In practice, in the most common cases, in classification a majority voting criterion is used, while in regression an empirical mean is used. In general, however, classification produces discrete labels, while regression produces continuous values.

Definition 5.2.1. In the k-Nearest Neighbors (KNN) regression given a value k and a prediction point x_0, the KNN regression identifies in the training set the k nearest observations, N_0

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_0} y_i \tag{39}$$

The optimal value of k is related to the trade-off variance-bias, in fact a small k leads to high variance and low bias, while a big k leads low variance (smoother prediction) and high bias and the local structure of $f(X)$ may not be captured.

Remark 5.2.2. Two important facts are:

- If the data has a clear shape, i.e. a line, the non parametric approach can perform poorly and it is preferred to use a parametric approach.
- By increasing the number of variables p, the KNN performance will rapidly decrease in terms of MSE test. It is more difficult to find the 'nearest neighbours' in higher dimension, this is called *curse of dimensionality.*

5.3 Local regression [1]

5.3.1 Basic formulation

We are interested in examining the relationship that links two quantities, represented by variables x and y, using a formula of the type

$$y = f(x) + \varepsilon \tag{40}$$

where ε is a random, non-observed error term. Without loss of generality, we can assume that $\mathbb{E}\{\varepsilon\} = 0$ because a possible nonzero value can be included in $f(x)$. This formulation is similar to that of (2.2) [1] in [1], but we do not presume that f is a member of a specific parametric class. We limit ourselves to looking for an estimate of $f(x)$, presuming only some regularity conditions.

Consider a general but fixed point x_0 of real numbers. We want to estimate $f(x)$ of 40 at point x_0.

If $f(x)$ is a derivable function with a continuous derivative at x_0, then, based on development of the Taylor series, $f(x)$ is locally approximated by a line passing through $(x_0, f(x_0))$, that is,

$$f(x) = \underbrace{f(x_0)}_{\beta_0} + \underbrace{f'(x_0)}_{\beta_1}(x - x_0) + \text{ remainder}$$

[1] More than a few times in this section references will be made to formulas in the book [1] by B. Scarpa. They are formulas related to linear regression, I think you can tell from the context, however I suggest going to the reference text. For simplicity, however, I report here the formulas (2.2) and (2.3) of [1]:

$$y = f(x; \beta) + \varepsilon \tag{2.2}$$

$$D(\beta) = \sum_{i=1}^{n} \{y_i - f(x_i; \beta)\}^2 = \|y - f(x; \beta)\|^2 \tag{2.3}$$

Nucleus	$w(z)$	Support
Normal	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}z^2\right)$	$\mathbb{R}$
Rectangular	$\frac{1}{2}$	$(-1,1)$
Epanechnikov	$\frac{3}{4}\left(1-z^2\right)$	$(-1,1)$
Biquadratic	$\frac{15}{16}\left(1-z^2\right)^2$	$(-1,1)$
Tricubic	$\frac{70}{81}\left(1-\lvert z\rvert^3\right)^3$	$(-1,1)$

Table 3: Some common Choices for Kernels

where the remainder is a quantity with an order of magnitude less than $|x-x_0|$. Transferring this idea to the context of statistical estimation, we estimate $f(x)$ in a neighborhood x_0 by means of a criterion that takes advantage of this fact, according to n observation pairs (x_i, y_i) for $i = 1, \ldots, n$. The remainder term is incorporated in ε.

Let us therefore introduce a criterion analogous to (2.3) in [1], but we now weigh observations based on their distance from x_0, which is

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} \left\{y_i - \beta_0 - \beta_1\left(x_i - x_0\right)\right\}^2 w_i \tag{41}$$

where weights w_i are chosen so that they are largest when $|x_i - x_0|$ is smallest. Formula 41 is a particular form of the *weighted least squares criterion*, a generalization of least squares when a set of weights is available. Following this criterion, the estimates of the parameters $\beta = (\beta_0, \beta_1)^\top$ are

$$\hat{\beta} = \left(X^\top W X\right)^{-1} X^\top W y$$

where X is a $n \times 2$ matrix whose i th row is $(1, (x_i - x_0))$, and W is the $n \times n$ diagonal matrix with w_i as diagonal elements. Because weights w_i are constructed with a "local" perspective around x_0, the resulting estimation method is called local regression. Minimization problem 41 is resolved by $\hat{\beta}$ and the estimate of $f\left(x_0\right)$ is $\hat{f}\left(x_0\right) = \hat{\beta}_0$. One way to select the weights is to set

$$w_i = \frac{1}{h} w\left(\frac{x_i - x_0}{h}\right)$$

where $w(\cdot)$ is a symmetric density function around the origin, which in this context, is called a kernel, and h (with $h > 0$) represents a scale factor, which is called bandwidth or smoothing parameter. Some of the more common choices for kernel $w(\cdot)$ are listed in Table 3. It is convenient to think of the normal kernel, corresponding to density $N(0,1)$, which we use from now on. Expression 41 depends on weights w_i, which in turn depend on elements h, $w(\cdot)$, and x_0. Even with h and kernel $w(\cdot)$ fixed, the minimization problem depends on x_0, and estimating $f(x)$ for different choices of x requires many minimization operations. Repeating the minimization operation is not a problem, as we can show that the estimate relative to a general point x can be obtained from the explicit formula:

$$\hat{f}(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{\{a_2(x;h)-a_1(x;h)\,(x_i-x)\}\,w_i y_i}{a_2(x;h)a_0(x;h)-a_1(x;h)^2} \tag{42}$$

where $a_r(x;h)=\{\sum(x_i-x)^r w_i\}/n$, for $r=0,1,2$. We are therefore dealing with an estimate that is noniterative and linear in the y_i, and can therefore write

$$\hat{f}(x)=s_h^\top y$$

for a suitable vector $s_h\in\mathbb{R}^n$ depending on h,x and $x_1,\ldots,x_n$. We do not usually estimate $f(x)$ at a single point, but on a whole set of m values (generally equally spaced) that span the interval of interest for variable x. We can calculate each of the m estimates by a single matrix operation of the type

$$\hat{f}(x)=S_h y \tag{43}$$

where S_h is an $m\times n$ matrix, called smoothing matrix; x is now the vector (in $\mathbb{R}^m$) of the x-axis where we estimate function f; and $\hat{f}(x)$ is the corresponding estimation vector.

If n is very large, we can reduce the size of matrix S_h by regrouping variable x into classes, and therefore use an $m\times n'$ matrix, with $n'\ll n$.

The choice to approximate a function $f(x)$ locally by a straight line may be relaxed by fitting a polynomial locally. Degree 0 and degree 2 are the alternatives in actual use. When a polynomial of degree 0 is used, the estimate of each point is a weighted mean of the data in a neighborhood of that point. However, a modification of this procedure with degree 0 , called k-nearest-neighbor and described later, is typically preferred. A polynomial of degree 2 is an appropriate choice when the data show sharp peaks and troughs, because this variant is more suitable for producing steep curves.

5.3.2 Choice of smoothing parameters

The problem of the choice of h and $w(\cdot)$ remains. The latter is not critical, as many studies on the subject have shown, and we can use any kernel listed in Table 3. At most, there is a slight benefit in using continuous functions and some computational advantages in the choice of kernels with limited support.

The truly important aspect is the choice of smoothing parameter h. Lowering value h produces curve $\hat{f}$, which is closer to the local behavior of the data and is therefore rougher, because the allocated weights system works on a smaller window and is more affected by local data variability. In the other direction, the increase in h produces the opposite effect:

the window on which the weights operate widens and the curve becomes smoother.

To understand which ingredients regulate the behavior of $\hat{f}$, particularly in relation to h, we must study the formal properties of $\hat{f}$. Limiting ourselves to quite simple working hypotheses, let us assume that var $\{\varepsilon_i\} = \sigma^2$ is a positive constant common to all observations and that the observations are not correlated. Under suitable regularity conditions for f, we can prove that for h sufficiently close to 0 and n sufficiently large, the approximations

$$\mathbb{E}\{\hat{f}(x)\} \approx f(x) + \frac{h^2}{2}\sigma_w^2 f''(x), \quad \text{var}\{\hat{f}(x)\} \approx \frac{\sigma^2}{nh}\frac{\alpha(w)}{g(x)} \tag{44}$$

hold, where $\sigma_w^2 = \int z^2 w(z) \mathrm{d}z, \alpha(w) = \int w(z)^2 \, \mathrm{d}z$, and $g(x)$ indicates the density from which the x_i were sampled.

These expressions show that bias is a multiple of h^2 and the variable is a multiple of $1/(nh)$. Therefore, although we would like to choose $h \to 0$ to bring down the bias, this makes the variance of the estimate diverge. For $h \to \infty$, the opposite occurs: the variance is reduced, but the bias diverges. Relations 44 are valid in the somewhat restrictive hypotheses previously mentioned, but the same type of indication is essentially valid with weaker hypotheses: the resulting formulas are more complex, but the qualitative indication is similar.

At this point, we can also verify the same contrast between the bias and variance of the estimate already seen in another context. As in that case, we must adopt a trade-off solution, balancing bias and variance in some way. In a certain sense, the optimal solution is implicit in relations 44. That is, minimizing the sum of the variance and the square of the bias, as indicated in Equation 35, the asymptotically best choice for h is

$$h_{\text{opt}} = \left(\frac{\alpha(w)}{\sigma_w^4 f''(x)^2 g(x) n}\right)^{1/5}. \tag{45}$$

However, this expression is not directly useful because it involves unknown terms $f''(x)$ and $g(x)$, although it does supply at least two important elements:

- it tells us that h must tend to 0 as $n^{-1/5}$, and therefore that it decreases very slowly;
- if we substitute this h_{opt} into the mean and variance expressions 44, it tells us that the mean squared error tends to 0 at a rate of $n^{-4/5}$; therefore this method of non-parametric estimation is intrinsically less efficient than a parametric one with a rate of decrease of n^{-1}, when the parametric model is satisfactory.

This last remark has much broader validity than is apparent here, in the sense that the basic indication is also valid for other methods of non-parametric estimation (see later).

Operatively, to choose h, we therefore take different routes to those in Equation 45, or at least we do not use it directly. A more formal process is the methods of cross-validation and AIC$_c$, which are in current use, having been suitably adapted to the problem. Specifically, the AIC$_c$ variant

$$\mathrm{AIC}_c = \log \hat{\sigma}^2 + 1 + \frac{2\left\{\mathrm{tr}\left(S_h\right) + 1\right\}}{n - \mathrm{tr}\left(S_h\right) - 2}$$

is proposed. Here

$$\hat{\sigma}^2 = \frac{1}{n}\sum_i \left(y_i - \hat{f}\left(x_i\right)\right)^2 = \frac{1}{n} y^\top \left(I_n - S_h\right)^\top \left(I_n - S_h\right) y$$

is the estimate of residual variance σ^2, and $\mathrm{tr}\left(S_h\right)$ indicates the trace of matrix S_h in 43. This trace is a substitutive measure of the number of parameters involved.

To conclude, we note that the linearity of the estimation process with respect to y_i, is valid when h is fixed independently of the data. However, if h is chosen on the basis of the same data, as commonly occurs, then the method is no longer linear.

5.3.3 Variability bands

To make inferences, it is useful to develop a tool that is similar to the confidence interval, to give the estimate of $f(x)$ an indicator of its reliability. To construct such an interval, we must refer to a pivotal quantity, at least approximately, of the type

$$\frac{\hat{f}(x) - f(x) - b(x)}{\sqrt{\mathrm{var}\{\hat{f}(x)\}}} \sim N(0,1) \tag{46}$$

where $b(x)$ indicates the bias of the estimate, of which the main term is approximated by the final term of the first expression of 44; analogously, the variance in the denominator is approximated by the second expression of 44. Note that for the asymptotically optimal bandwidth 45, the bias has the same order of magnitude as the denominator of 46. Therefore, the bias term cannot be neglected in this framework, in contrast with what happens in a parametric context.

Of the various quantities in play, all, in some way, can be computed at least approximately, except term $f''(x)$, which is included in bias $b(x)$. This means that constructing a confidence interval is not feasible, even in an approximate form. Instead of looking for extremely complicated corrections to remedy the problem, a current solution is to construct variability bands of the type

$$\left(\hat{f}(x) - z_{\alpha/2}\,\mathrm{std}\cdot\mathrm{err}(\hat{f}(x)), \hat{f}(x) + z_{\alpha/2}\,\mathrm{std}\cdot\mathrm{err}(\hat{f}(x))\right)$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of distribution $N(0,1)$ and $\mathrm{std}\cdot\mathrm{err}(\hat{f}(x))$ the denominator of 46. Strictly speaking, the previous expression is clearly that of an interval, but once the expression is applied to every point on the x-axis, it gives rise to two bands.

Remark 5.3.1. Two observations are necessary:

1. for every fixed x, the previous interval does not constitute a confidence interval, for the reasons already mentioned, but only provides an indication of the local variability of the estimate;

2. even if bias $b(x)$ were not present, the interval thus constructed would have a confidence level of approximately $1-\alpha$ for $f(x)$ to each fixed value of x, but not globally for the entire curve.

5.3.4 Variable bandwidths and loess

There are several variations to the basic method of local regression as described up to now. The most common variation regards the use of a non-constant bandwidth along the x-axis, but according to the level of sparseness of observed points.

These intuitive considerations are confirmed by expression 45, in which the presence of $g(x)$ in the denominator shows that when density $g(x)$ is low, that is, when observations x_i are sparse, we must use a larger value of h to keep $\mathrm{var}\{\hat{f}(x)\}$ the same.

One technique, which arose from these considerations, is loess (locally weighted scatterplot smoothing), which is very similar to the local regression. A distinctive feature of loess is that it expresses the smoothing parameter by means of the fraction of effective observations for estimating $f(x)$ at a certain point on the x-axis; this fraction is kept constant. Loess widens or narrows the window, so that the fraction of observations involved remains constant.

We can now see that the degree of smoothing is regulated by the fraction of points used, just like the bandwidth. Therefore, this fraction constitutes the smoothing parameter in loess.

Another typical aspect of loess is that it combines the ideas of local regression and robust estimation, which means that we substitute the quadratic function of 41 with another objective function that limits the effect of anomalous observations, commonly called outliers.

Again according to the robustness considerations of the procedure, loess uses a limited support kernel, generally tricubic (see table 3), which also has the advantage of more clearly distinguishing between used and unused points in the estimate.

The local regression of degree 0, when a non-constant bandwidth along the x-axis is chosen, is very simple and quite commonly used. The estimate of the function at each point is obtained as the average of a fixed number of closest observations around that point. This method is called k-nearest-neighbor, where k denotes the number of observations averaged by the estimate. We use k to indicate the decreasing complexity of the procedure because, when $k = n$, the estimate is simply the average of all available observations, giving a constant fit over the entire x-axis. Instead, when $k = 1$, the value of y of the closest observation is used at every single point as an estimate of the function, producing a very rough curve.

5.3.5 Extension to Several Dimensions

The formulation of section before may also be applied when two or more covariates, say, p, are used. Let us begin with the simplest case of two variables, x_1 and x_2, and presume that a relationship of the type

$$y = f(x_1, x_2) + \varepsilon$$

holds, where $f(x_1, x_2)$ is now a functibn from $\mathbb{R}^2$ to $\mathbb{R}$. The available data are now made up of the same y_i as previously and of points $x_i = (x_{i1}, x_{i2}) \in \mathbb{R}^2$, for $i = 1, \ldots, n$. To estimate f corresponding to a specific point, $x_0 = (x_{01}, x_{02})$, a natural extension of the criterion 41 takes the form

$$\min_{\beta_0,\beta_1,\beta_2} \sum_{i=1}^{n} \left\{y_i - \beta_0 - \beta_1 (x_{i1} - x_{01}) - \beta_2 (x_{i2} - x_{02})\right\}^2 w_i \tag{47}$$

where weights w_i are now to be determined as a function of a suitable distance between x_i and x_0. A common way of choosing w_i is to set

$$w_i = \frac{1}{h_1 h_2} w\left(\frac{x_{i1} - x_{01}}{h_1}\right) w\left(\frac{x_{i2} - x_{02}}{h_2}\right)$$

which is a simple extension of what we saw in local regression. Clearly, this expression involves two smoothing parameters, h_1 and h_2, to take into account the different variability of x_1 and x_2.

From a computational point of view, we can also tackle this problem as a variation of weighted least squares. If we indicate by X the $n \times 3$ matrix of which the i-th row is

$$(1, x_{i1} - x_{01}, x_{i2} - x_{02})$$

$y = (y_1, \ldots, y_n)^\top$ and $W = \operatorname{diag}(w_1, \ldots, w_n)$, then the solution of the previous minimum problem is the first element, which corresponds to β_0, of $\left(X^\top W X\right)^{-1} X^\top W y$. Obviously, this calculation is repeated for every choice of point x_0, and tendentially the number of these points is now higher than in the scalar case of the local regression.

Formally, most of the results can be easily extended to the multivariate case, where the formulation is of the type

$$y = f(x) + \varepsilon = f(x_1, \ldots, x_p) + \varepsilon \tag{48}$$

Definition of the estimation method seen for $p = 1$ and $p = 2$ extends naturally to the case of general p, meaning that there is no need to repeat the discussion of various connected aspects, such as the choice of h, and so on.

5.3.6 Curse of dimensionality

In practice, we rarely go much beyond two dimensions in non-parametric regression. The first reason is the poor conceptual manageability of the resulting object: although the idea of a function with 6 or 26 variables is not conceptually different from one with 2 variables, it is actually difficult to visualize mentally and graphically. Interpreting the results is also difficult.

A second and perhaps more important aspect is that with increasing dimension p of the space in which the covariates are placed, the observed points scatter very quickly. To understand the essence of the problem intuitively, think of $n = 500$ points on the x-axis, randomly set over an interval that, without loss of generality we may presume to be unit interval $(0, 1)$. If we use these n points to estimate function $f(x)$, we obtain a reliable estimate, thanks to the small average distance that separates them. If the same number n of points is then distributed in square $(0, 1)^2$ of plane (x_1, x_2), they are much less close to each other. If we then move to higher dimensions, say, p, the dispersion of n points in space $\mathbb{R}^p$ increases very rapidly, and the quality of the obtainable estimate correspondingly worsens.

To compensate for the increased space between the points, we need a number of points of the order of magnitude n^p. However, although it is common to use a sample of size $n = 500$, it is much more uncommon to have 500^5 units available, and practically impossible to have 500^{10}, even in a data mining context. These are the sizes that are in some way equivalent to estimating function f nonparametrically when the number of covariates is 5 or, respectively, 10.

This situation of substantial impossibility in estimating function f accurately when p is large is called the curse of dimensionality. For a more detailed explanation of how the scatter of the points increases with p, and for other similar issues, see Hastie et al. (2009; section 2.5).

A further critical aspect with increasing p is the increased computational cost, at least when a substantial increase in n also occurs.

These problems are not confined to the specific technique of local regression, but they are substantially valid for all nonparametric estimation techniques, as they are due to the dimension and dispersed nature of the data with respect to the number of points from which we wish to estimate the function and not so much to the method chosen for data processing.

To overcome the problem of the curse of dimensionality, one strategy is to carry out a preliminary operation to reduce the number p of the covariates, transforming them into a reduced set of new variables but at the same time losing as little of their informative content as possible.

The simplest and probably most frequent way of achieving this is to extract some of the principal components of the original covariates. Therefore, once the complete set of principal components has been constructed, a suitable number of them are chosen, keeping a sufficient proportion of the original variability and the number of new variables low. For a discussion of the advantages and disadvantages of PCA, see section 3.6.2 of [1].

Therefore, in the following section, what we indicate as covariates may not represent the original variables but those constructed through principal components or other methods of dimensionality reduction.

5.4 Splines [1]

The term spline originally meant the flexible strips of wood used to shape ships' hulls. Some points on the cross-section of the hull were chosen, and the rest of the curve of the hull was derived by forcing the wooden strips to pass through such points, leaving them free to fit into the rest of desired curve according to their natural tendency. This gave rise to a regular curve with preassigned behavior in certain positions.

5.4.1 Spline functions

The term spline is used in mathematics to construct piecewise polynomial functions, according to a logic that partly replicates the mechanism just described, to approximate functions of which we know the value only at certain points.

We choose K points $\xi_1 < \xi_2 < \cdots < \xi_K$, called knots, along the x-axis. A function $f(x)$ is constructed so that it passes exactly through the knots and is free at the other points, with the constraint that it presents regular overall behavior. In this sense, the function behaves like splines used in shipyards.

The following strategy is followed: between two successive knots, say, in the interval (ξ_i, ξ_{i+1}), curve $f(x)$ coincides with a suitable polynomial, of prefixed degree d, and these sections of polynomials meet at point $\xi_i(i = 2, \ldots, K-1)$, in the sense that the resulting function $f(x)$ has a continuous derivative from degree 0 to degree $d-1$ in each of the ξ_i.

The degree that is almost universally used is $d = 3$, and we therefore speak of cubic splines. The reason for this is that the human eye cannot perceive discontinuity in the third derivative. The foregoing conditions are therefore written as

$$
\begin{aligned}
&f\left(\xi_i\right) = y_i, \quad \text{for } i = 1, \ldots, K \\
&f\left(\xi_i^-\right) = f\left(\xi_i^+\right), \quad f'\left(\xi_i^-\right) = f'\left(\xi_i^+\right), \quad f''\left(\xi_i^-\right) = f''\left(\xi_i^+\right), \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{for } i = 2, \ldots, K-1
\end{aligned}
$$

. where $g\left(x^-\right)$and $g\left(x^+\right)$indicate the left and right limits of a function $g(\cdot)$ at point x.

The framework of the problem requires the following set of conditions: each of the $K-1$ cubic components requires four parameters; there are K constraints of the type $f\left(\xi_i\right) = y_i$, and $3(K-2)$ continuity constraints of the function and the first two derivatives.

As the difference between coefficients and constraints is 2 units, the system of conditions does not univocally identify a function. We must therefore introduce two additional constraints.

Many proposals have been made to define these additional constraints, most of which concern the outmost interval or the extreme points of the function. A particularly simple choice consists of constraining the second derivatives of the polynomials in the two extreme intervals to $0, f''\left(\xi_1\right) = f''\left(\xi_K\right) = 0$, which means that the two extreme polynomials are straight lines. The resulting function $f(x)$ is called the natural cubic spline.

5.4.2 Regression splines

The previous tool is also useful in statistics, in various forms, in the study of relations between a covariate x and a response y, for which we use n pairs of observations (x_i, y_i) for $i = 1, \ldots, n$.

Let us begin by applying these ideas to parametric regression. We return to model (2.2) of [1], where $f(x; \beta)$ is hypothesized to be a spline function. Then we divide the x-axis into $K+1$ intervals separated by K knots, $\xi_1, \ldots, \xi_K$, and interpolate the n points with criterion (2.3) of [1], where the β coefficients are now the non-constrained parameters of the $K+1$ constituent polynomials.

With respect to section 5.4.1, there is a certain difference in that the spline function coefficients can no longer be chosen according to constraints of the type $f\left(\xi_j\right) = y_j$, because K and n are no longer linked and $K \ll n$. This means that we have to use a fitting criterion between the data and the interpolated function, for example, the least squares criterion or a similar one.

If we use cubic splines, the total number of cubic parameters is $4(K+1)$ subject to $3K$ continuity constraints, and therefore β has $K+4$ components. The solution to the minimum problem (2.3) of [1] may be rewritten in the equivalent form

$$f(x;\beta) = \sum_{j=1}^{K+4} \hat{\beta}_j h_j(x) \tag{49}$$

where

$$\begin{aligned} h_j(x) &= x^{j-1} \quad \text{for } j = 1, \ldots, 4 \\ h_{j+4}(x) &= (x - \xi_j)_+^3, \quad \text{for } j = 1, \ldots, K \end{aligned}$$

and $a_+ = \max(a, 0)$. The solution is thus represented by a suitable linear combination of basis functions $\{h_j(x), j = 1, \ldots, K+4\}$, composed partly of low-order powers of x and partly of functions of the type $\max\left(0, (x-\xi)^3\right)$.

The number K of knots and their position along the x-axis need to be chosen. Because K is viewed as a tuning parameter, regulating the complexity of the model, the strategies of bias-variance tradeoff or traninig-validation set, apply. Once K has been set, when no information is available about the shape of the function to be estimated, a reasonable choice for knot positions is uniformly along the x_i range. Alternatively, the quantiles of the empirical distribution of the x_i are chosen as knots.

For $d = 1$, the basis is represented by

$$h_1(x) = 1, \quad h_2 = x, \quad h_{j+2}(x) = (x - \xi_j)_+, \quad \text{for } j = 1, \ldots, K$$

5.4.3 Smoothing splines

Another way of using spline functions in studying the relationship between variables is to introduce an approach to non-parametric estimation as an alternative to local regression. Let us consider the penalized least squares criterion

$$D(f, \lambda) = \sum_{i=1}^{n} \left[y_i - f(x_i)\right]^2 + \lambda \int_{-\infty}^{\infty} \{f''(t)\}^2 \, \mathrm{d}t \tag{50}$$

where λ is a positive penalization parameter of the roughness degree of curve f, quantified by the integral of $f''(x)^2$, and therefore acts as a smoothing parameter. If $\lambda \to 0$, there is no penalization for the roughness of $f(x)$, so the previous criterion is not influenced by $f(x)$ outside points $x_1, \ldots, x_n$, and the optimal solution $\hat{f}(x_i)$ is the arithmetic mean of the y_i corresponding to each fixed x for each of the observed x_i but is not determined for other values of x. If $\lambda \to \infty$, the penalty is maximal and means adapting a line imposing $f''(x) \equiv 0$. The overall result is the least squares line.

Therefore, the role of λ is qualitatively similar to that of h in the case of local regression.

A noteworthy mathematical result (Green & Silverman 1994) shows that the solution to the minimization problem 50 is represented by a natural cubic spline, whose knots are distinct points x_i. The solution may be written as

$$\hat{f}(x) = \sum_{j=1}^{n_0} \theta_j N_j(x)$$

where n_0 is the number of distinct x_i and the $N_j(x)$ are natural cubic splines basis functions. We can rewrite

$$D(f, \lambda) = (y - N\theta)^\top (y - N\theta) + \lambda \theta^\top \Omega \theta$$

where N is the matrix in which the j th column contains the values of N_j corresponding to the n_0 distinct values of x, and Ω is the matrix of which the generic element is $\int N_j''(t) N_k''(t) \mathrm{d}t$. The solution of the optimization problem is given by

$$\hat{\theta} = \left(N^\top N + \lambda \Omega\right)^{-1} N^\top y \tag{51}$$

which clearly depends on the choice of smoothing parameter λ. If this expression of $\hat{\theta}$ is substituted into that of $f(x)$, we have $\hat{y} = \tilde{S}_\lambda y$ for a certain matrix $\tilde{S}_\lambda$ of dimension $n_0 \times n_0$, that is, we are dealing with another linear smoother. In this case, we speak of smoothing splines.

However, from a computational point of view, we do not proceed with 51, which involves a matrix of order n_0. There are much more efficient algorithms, for which we refer readers to the specialized literature (see the bibliographical notes). In addition, when the quantity of data is very large, we can reduce the number of knots used, without loss of accuracy, as we did for local regression.

5.4.4 Additive models and generalized addivite model [1]

Up to now, we have examined various methods of non-parametric regression estimation, each of which allows us to examine the relationship between a response variable y and a certain number p of explanatory variables. All these techniques are valid for the aim, but they also come up against the same problems when p is high: the curse of dimensionality and the other aspects discussed before.

To overcome this, on one hand we have to introduce some form of "structure", that is, a model of the form of regression function $f(x), x = (x_1, \ldots, x_p) \in \mathbb{R}^p$. On the other hand, for reasons already discussed, we do not want a rigid structure but must maintain ample flexibility.

One option that has been greatly appreciated for its practical usefulness and logical simplicity is the following. Let us presume that a representation of the type

$$f(x)=f\left(x_1, \ldots, x_p\right)=\beta_0+\sum_{j=1}^p f_j\left(x_j\right) \tag{52}$$

holds for $f(x)$, where $f_1, \ldots, f_p$ are functions of one variable, each having smooth behavior, and β_0 is a constant. We say that formulation 48 with representation 52 of $f(x)$ is an additive model.

Note that to avoid what is essentially a problem of model identifiability, it is necessary for the various f_j to be centred around 0 , that is,

$$\sum_{i=1}^n f_j\left(x_{ij}\right)=0, \quad(j=1, \ldots, p)$$

where x_{ij} is the j the variable for unit i.

To fit 52 to the data, there is an iterative process based on a non-parametric estimation method of one-variable functions to estimate f_j. This procedure, shown in the next algorithm, is called *backfitting* and is essentially a variation of the Gauss-Seidel algorithm.

```
1. Start: $\hat{\beta}_0 \leftarrow \sum_i y_i/n, \hat{f}_j \leftarrow 0$ for all $j$.
2. Cycle for $j=1,2,\ldots,p,1,2,\ldots,p,1,2,\ldots$ :
     a $\hat{f}_j \leftarrow \mathcal{S}\left[\left\{y_i-\hat{\beta}_0-\sum_{k \neq j} \hat{f}_k\left(x_{ik}\right)\right\}_1^n\right]$,
     b $\hat{f}_j \leftarrow \hat{f}_j-n^{-1} \sum_{i=1}^n \hat{f}_j\left(x_{ij}\right)$, until functions $\hat{f}_j$ stabilize.
```

The specific method for non-parametric estimation is not crucial, and we can even choose different methods for different f_j, but we usually apply a single one, generically indicated by $\mathcal{S}$ in the algorithm, in the sense that $\mathcal{S}(y)$ constitutes the non-parametric estimate, calculated on the observed values $y=\left(y_1, \ldots, y_n\right)^{\top}$, of a scalar function. In many cases, $\mathcal{S}$ is a linear estimator, of type Sy, where S is a suitable smoothing matrix. A generalization of model 52 is of the type

$$\begin{aligned}
f\left(x_1, \ldots, x_p\right)= & \beta_0+\sum_{j=1}^p f_j\left(x_j\right)+\sum_{j=1}^p \sum_{k<j} f_{kj}\left(x_k, x_j\right) \\
& +\sum_{j=1}^p \sum_{k<j} \sum_{h<k<j} f_{hkj}\left(x_h, x_k, x_j\right)+\cdots
\end{aligned}$$

which allows us to bear in mind the interaction effect between pairs of variables, triplets, or other interactions of a higher order. Another direction in which model 52 is frequently generalized is of the type

$$g\left(\mathbb{E}\left\{Y \mid x_1, \ldots, x_p\right\}\right)=\beta_0+\sum_{j=1}^p f_j\left(x_j\right)$$

which is called *generalized additive model* (GAM). As in the standard GLM, link function g must be specified. For example, in the case of binomial Y, g is commonly assumed to be logit function. Instead, the term on the right-hand side is now expressed by an additive form, and consequently the contribution of general variable x_j is no longer linear $\beta_j x_j$ but is of the more general type $f_j(x_j)$.

To estimate functions for a GAM-type model, we use a suitable combination of the algorithm with that of iterative weighted least squares, applied in the case of GLM.

5.5 Generalized Additive Models [5]

So far we have seen a number of approaches for flexibly predicting a response y on the basis of a single predictor x. These approaches may be seen as extensions of simple linear regression. Here we explore the problem of flexibly predicting y on the basis of several predictors, $x_1, \ldots, x_p$. *Generalized additive models* (GAMs) provide a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity. The beauty of GAMs is that we can use splines and local regression as building blocks for fitting an additive model. A natural way of extending the multiple linear regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

in order to allow for nonlinear relationships between each feature and the response is to replace each linear component $\beta_j x_{ij}$ with a smooth nonlinear function $f_j(x_{ij})$.

Definition 5.5.1. If we do so we obtain the *Generalized Additive Model* (GAM)

$$y_i = \beta_0 + \sum_{i=j}^{p} f_j(x_{ij}) + \varepsilon_i = \beta_0 + f_1(x_{i1}) + f_2(x_i 2) + \cdots + f_p(x_i p) + \varepsilon_i \quad (53)$$

It is called additive because we calculate a separate f_j for each X_j and then add together all of their contributions.

Remark 5.5.2. GAM has some important properties:

1. GAMs allow us to fit a non-linear f_j to each x_j, so that we can automatically model non-linear relationships that standard linear regression will miss. This means that we do not need to manually try out many different transformations on each variable individually.

2. The non-linear fits can potentially make more accurate predictions for the response y.

3. Because the model is additive, we can still examine the effect of each x_j on y individually while holding all of the other variables fixed.

4. If we are interested in inference, GAMs provide a useful representation.

5.6 Regression trees [1]

5.6.1 Approximations via step functions

In one sense, the simplest way to approximate a generic function $y = f(x)$, with $x \in \mathbb{R}$, is to use a step function, that is, a piecewise constant function.

However, there are various choices to be made:

a how many subdivisions of the x-axis must be considered?

b where are the subdivision points to be placed?

c which value of y must be assigned to each interval?

Of these questions, the easiest to answer is the last one, because it is completely natural to choose value $\int_{R_j} f(x)\mathrm{d}x/|R_j|$ for any interval R_j, having indicated the length of that interval by $|R_j|$. Regarding positioning the subdivision points of $\mathbb{R}$, and therefore defining the intervals, it is better to choose small intervals where $f(x)$ is steeper. The choice of the number of subdivisions is the most subjective of the three points: intuitively, any increase in the number of steps increases the quality of the approximation, and therefore, in a certain sense, we are led to think of infinite subdivisions. However, this is counter to the requirement to use a ”sparing” approximate representation, and therefore to adopt a finite number of subdivisions.

The scheme can be extended to the case of functions of p variables: we thus write $y = f(x)$ where $x \in \mathbb{R}^p$. There are many ways of extending the idea from the $p = 1$ case to the general p case.

The components of the tree are inequalities, called nodes, relative to any component x. We begin by examining the inequality of the node at the root of the tree, which is at the top. We follow the left branch if the inequality is true and the right branch if it is not. We proceed in the same way, sequentially examining all the inequalities until we reach the terminal nodes, called leaves, which give the values of the approximating function.

The tree is identified by a few numerical elements, it can easily be stored. A second important advantage is that we can move from one approximation to a more accurate one by subdividing one of the components into two subrectangles with the same characteristics as the original. This corresponds to extending a branch of the tree to a further branch level. This characteristic immediately allows us to recursively construct a sequence of approximations that are increasingly accurate, each obtained by refining the previous one.

5.6.2 Regression trees: growth

We want to use the idea of approximation with a step function to approximate our functions of interest, which are regression functions. Obviously, in our context, regression function $f(x)$ is not known, but we can observe it indirectly through n sample observations, generated by model 48.

For simplicity, we begin from the case where $p = 1$ and we can estimate regression curve $f(x)$ underlying the data by a step function of the type just described, that is

$$\hat{f}(x) = \sum_{h=1}^{J} c_h I\left(x \in R_h\right) \tag{54}$$

regression function $f(x)$ is not known, but we can observe it indirectly through n sample observations, generated by model 48.

A crucial aspect is the fact that at each step, one of the already constructed rectangles is divided into two, and so is the portion of data belonging to it; we optimize deviance with respect to this operation. Therefore, this is a myopic optimization procedure. Although it does not guarantee global minimization of deviance, it does provide acceptable solutions, maintaining limited computational complexity. At least in principle, this procedure can be applied iteratively through successive subdivisions of $\mathbb{R}^p$ until we can no longer distinguish sets containing a single sampled observation and thus obtain a tree with n leaves. To be useful, the number of leaves must be less than n, preferably much less. Therefore, after the stage of tree growth, with the complete or almost complete development of all the leaves, we move to a stage of tree pruning. We describe the growth algorithm now and return to the pruning phase later.

To develop the growth algorithm, first note that whatever the division of $\mathbb{R}^p$ into hyper-rectangles, we can break down the deviance as follows

$$D = \sum_{i=1}^{n} \left\{ y_i - \hat{f}\left(x_i\right) \right\}^2 = \sum_{h=1}^{J} \left\{ \sum_{i \in R_h} \left(y_i - \hat{c}_h\right)^2 \right\} = \sum_h D_h. \tag{55}$$

We also bear in mind the general property that the minimum of $\sum_{i=1}^{n} \left(z_i - a\right)^2$ with respect to a is obtained for $a = M(z)$, where $M(\cdot)$ is the average operator of the vector.

The growth process starts with $J = 1, R_J = \mathbb{R}^p, D = \sum_i \left(y_i - M(y)\right)^2$, and proceeds iteratively for a number of cycles, according to the following scheme: - once a rectangle R_h is chosen, the appropriate value c_h is the average of the corresponding values

$$\hat{c}_h = M\left(y_i : x_i \in R_h\right)$$

- if we subdivide region R_h into two parts, R'_h and R''_h (therefore moving to $J+1$ leaves), summand D_h of D is replaced by

$$D^*_h = \sum_{i \in R'_h} \left(y_i - \hat{c}'_h\right)^2 + \sum_{i \in R''_h} \left(y_i - \hat{c}''_h\right)^2$$

with a "gain" of

$$g_h = D_h - D^*_h \tag{56}$$

- we can inspect all p explanatory variables and, for each of them, all the possible points of subdivision, selecting the variable and its point of subdivision that maximize g_h.

We stop when $J = n$, at least conceptually. Mainly, if n is enormous, we stop earlier - for example, when all the leaves contain a number of sample elements that is less than a preassigned value, or when the relative fall of deviance is less than a prefixed threshold.

6 Gradient Boosting [5], [9]

The idea of gradient boosting for classification is to assign more weight to observations badly classified, to make the model work more on these (AdaBoost). Let's give a first intuition on what is gradient boosting. Let us consider a simple regression problem, with a simple case:

$$\left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_n, y_n\right)$$

. We want to estimate a model $y = f(x)$ minimizing a loss function, i.e. Mean Squared Error. Suppose that we have a good model f, but we notice some errors: $f\left(x_1\right) = 0.8$ while $y_1 = 0.9, f\left(x_2\right) = 1.4$ while $y_2 = 1.3$. How can we improve the model? We can not modify f but we can add to f another model, such as regression tree, h, so that the new prediction will be

$$y_i = f\left(x_i\right) + h\left(x_i\right) \tag{57}$$

Hence the prediction is updated as follows:

$$\begin{gathered} f\left(x_1\right) + h\left(x_1\right) = y_1 \\ f\left(x_2\right) + h\left(x_2\right) = y_2 \\ \vdots \\ f\left(x_n\right) + h\left(x_n\right) = y_n . \end{gathered}$$

But we can also write

$$\begin{aligned} y_1 - f(x_1) &= h(x_1) \\ y_2 - f(x_2) &= h(x_2) \\ &\vdots \\ y_n - f(x_n) &= h(x_n) \end{aligned}$$

where $r(x_i) = y_i - f(x_i)$ are the residuals. Gradient Boosting fits a regression tree, h, on data $(x_1, r_1), (x_2, r_2), \dots, (x_n, r_n)$ to improve the prediction. The role of h is to compensate the 'problems' of model f.

So we have a new model for y, which should be better than the previous one:

$$f_2(x) = f_1(x) + h_1(x)$$

and we can repeat this reasoning obtaining the residuals with respect to this new model $f_2(\cdot)$ and fit a new tree $h_2(x_i)$ to further improve the prediction. Thus the prediction will be

$$f_3(x) = f_2(x) + h_2(x)$$

We can repeat this M times and at each iteration $1 < m < M$ we will have

$$f_{m+1}(x) = f_m(x) + h_m(x)$$

How is this related to the Gradient Descent? Let us consider the quadratic loss function

$$L(y, f(x)) = \frac{1}{2}(y - f(x))^2$$

We want to minimize $J = \sum_i L(y_i, f(x_i))$

$$\frac{\partial J}{\partial f(x_i)} = \frac{\partial \sum_i L(y_i, f(x_i))}{\partial f(x_i)} = \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = f(x_i) - y_i$$

We can see the residuals as negative gradients

$$-g(x_i) = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right] = y_i - f(x_i)$$

Gradient Descent minimizes a function going in the opposite direction with respect to the gradient

$$\vartheta_{m+1} = \vartheta_m - \rho \frac{\partial J}{\partial \vartheta_m}$$

For a regression problem with quadratic loss function:

1. residual ↔ negative gradient
2. fit h to the residual ↔ fit h to the negative gradient;
3. update f through the residual ↔ update f through the negative gradient

We are using the negative gradient.

6.1 Step function

In one sense, the simplest way to approximate a generic function $f(x)$ is to use a step function, that is, a piecewise constant function. The previous scheme can be extended to the case of functions $f(x)$ of p variables, $x = (x_1, \ldots, x_p)$. To keep simple the method we require that the regions with constant values are rectangles, the sides of which are parallel to the coordinate axes. This approximate function may be represented as a binary tree

6.2 Regression tree

We want to estimate regression curve $f(x)$ underlying the data by

$$\hat{f}(x) = \sum_{j=1}^{J} c_j I\left(x \in R_j\right) \tag{58}$$

where $I(x \in A)$ is the indicator function of the set A (and here they are rectangles) and $c_1, \ldots, c_J$ are constants objective function: deviance,

$$D = \sum_{i} \left\{ y_i - \hat{f}\left(x_i\right) \right\}^2 \tag{59}$$

This minimization, even if we fix the number of steps J, involves very complex computation. Operatively we follow a sub-optimal approach of step-by-step optimization: we construct a sequence of gradually more refined approximations and to each of these we minimize the deviance relative to the passage from the current approximation to the previous one. It is not ensured that we get the global maximum. This procedure is called greedy-algorithm or myopic optimization. This operation is represented by a series of binary splits. Each internal node represents a value query on one of the variables - e.g. 'Is $x_3 > 0.4$?'. If the answer is 'Yes', go right, else go left. The terminal nodes are the decision nodes. Typically each terminal node is assigned a value, c_h, given by the arithmetic mean of the observed y_i having component x_j falling in this node.

6.3 Gradient Boosting: Algorithm

A Gradient Boosting may be defined with these input elements: a training set $(x_i, y_i) \ldots (x_n, y_n)$, a loss function $L(y, f(x))$, the number of iterations M.Gradient Tree Boosting algorithm is following:

1. initialize the model with a constant value

$$f_0(x) = \arg\min_{\gamma} \frac{1}{n} \sum_{i=1}^{n} L\left(y_i, \gamma\right) \tag{60}$$

with quadratic loss function we have $f_0 = \bar{y}$

2. at each iteration $1 < m < M$ calculate the negative gradients for $i = 1, 2, \ldots, n$
$$-g\left(x_i\right) = -\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)} \tag{61}$$

3. estimate a regression tree $h_m(x)$ on $-g\left(x_i\right)$ giving terminal regions $R_{jm}, j = 1, 2, \ldots, J_m$

4. for $j = 1, 2, \ldots, J_m$ calculate $\gamma_{jm} = \arg\min \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}\left(x_i\right) + \gamma\right)$

5. update the model $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I\left(x \in R_{jm}\right)$

6. Output: $\hat{f}(x) = f_M(x)$

We use the negative gradients because we can use loss functions other than the quadratic loss and derive the corresponding algorithms.

Remark 6.3.1. Quadratic loss function is simple to handle mathematically but not robust with respect to outliers and the presence of an outlier may have negative effects on the general performance of the model. Other loss functions are:

- absolute loss function
$$L(y, f) = |y - f| \tag{62}$$

- Huber loss function, which is more robust with respect to outliers
$$L(y, f) = \begin{cases} \frac{1}{2}(y - f)^2, & |y - f| \leq \delta \\ \delta(|y - f| - \frac{1}{2}\delta), & |y - f| > \delta \end{cases} \tag{63}$$

6.4 Gradient Boosting: regularization

As in other models, also in the case of the Gradient Boosting we can introduce some regularization techniques, in order to reduce the risk of overfitting. The update rule is modified in this way

$$f_m(x) = f_{m-1}(x) + \nu \cdot \sum_{j=1}^{J} \gamma_{jm} I\left(x \in R_{jm}\right) \tag{64}$$

Parameter $0 < \nu < 1$ controls the 'learning rate' of the boosting procedure. Smaller values of ν implies more shrinkage, which implies a bigger M. So we have a trade-off between ν and M. At the end gradient boosting is great for use of 'mixed' data, it is robust to outliers in input, has a good predictive power and its results are interpretable.

7 Appendices

7.1 ARIMA in details [2]

7.1.1 Definitions

Definition 7.1.1 (Autoregressive Model)**.** An AR(p) (AutoRegressive of order p) model is a discrete time linear equations with noise, of the form:

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \varepsilon_t \tag{65}$$

Here p is the order, $\alpha_1, \ldots, \alpha_p$ are the parameters or coefficients (real numbers), ε_t is an error term, usually a white noise with intensity σ^2. The model is considered either on integers $t \in \mathbb{Z}$, thus without initial conditions, or on the non-negative integers $t \in \mathbb{N}$. In this case, the relation above starts from $t = p$ and some initial condition $X_0, \ldots, X_{p-1}$ must be specified.

Example 7.1.2. We have seen above the simplest case of an $AR(1)$ model

$$X_t = \alpha X_{t-1} + \varepsilon_t$$

With $|\alpha| < 1$ and $\operatorname{Var}[X_t] = \frac{\sigma^2}{1-\alpha^2}$, it is a wide sense stationary process (in fact strict sense since it is gaussian). The autocorrelation coefficient decays exponentially:

$$\rho(n) = \alpha^n.$$

Even if the general formula is not so simple, one can prove a similar result for any AR model.

In order to model more general situations, it may be convenient to introduce models with non-zero average, namely of the form

$$(X_t - \mu) = \alpha_1 (X_{t-1} - \mu) + \ldots + \alpha_p (X_{t-p} - \mu) + \varepsilon_t.$$

When $\mu = 0$, if we take an initial condition having zero average (this is needed if we want stationarity), then $E[X_t] = 0$ for all t. We may escape this restriction by taking $\mu \neq 0$. The new process $Z_t = X_t - \mu$ has zero average and satisfies the usual equation

$$Z_t = \alpha_1 Z_{t-1} + \ldots + \alpha_p Z_{t-p} + \varepsilon_t$$

But X_t satisfies

$$\begin{aligned} X_t &= \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \varepsilon_t + (\mu - \alpha_1 \mu - \ldots - \alpha_p \mu) \\ &= \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \varepsilon_t + \widetilde{\mu}. \end{aligned}$$

Definition 7.1.3 (Lag operator)**.**

Let S be the space of all sequences $(x_t)_{t\in\mathbb{Z}}$ of real numbers. Let us define the Time lag operator $L : S \to S$, a map which transform sequences in sequences. It is defined as

$$Lx_t = x_{t-1}, \qquad \text{for all } t \in \mathbb{Z}.$$

We should write $(Lx)_t = x_{t-1}$, with the meaning that, given a sequence $x = (x_t)_{t\in\mathbb{Z}} \in S$, we introduce a new sequence $Lx \in S$, that at time t is equal to the original sequence at time $t-1$, hence the notation $(Lx)_t = x_{t-1}$. For shortness, we drop the bracket and write $Lx_t = x_{t-1}$, but it is clear that L operates on the full sequence x, not on the single value x_t.

The map L is called time lag operator, or backward shift, because the result of L is a shift, a translation, of the sequence, backwards (in the sense that we observe the same sequence but from one position shifted on the left).

If we work on the space S^+ of sequences $(x_t)_{t\in\mathbb{N}}$ defined only for non-negative times, we cannot define this operator since, given $(x_t)_{t\in\mathbb{N}}$, its first value is x_0, while the first value of Lx should be x_{-1} which does not exist. Nevertheless, if we forget the first value, the operation of backward shift is meaningful also here. Hence the notation $Lx_t = x_{t-1}$ is used also for sequences $(x_t)_{t\in\mathbb{N}}$, with the understanding that one cannot take $t = 0$ in it.

Remark 7.1.4. The time lag operator is a linear operator.

The powers, positive and negative, of the lag operator are denoted by L^k :

$$L^k x_t = x_{t-k}, \qquad \text{for } t \in \mathbb{Z}$$

(or, for $t \geq \max(k, 0)$, for sequences $(x_t)_{t\in\mathbb{N}}$). With this notation, the AR model reads

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right) X_t = \varepsilon_t$$

Definition 7.1.5 (Moving Average Model)**.** AMA(q) (Moving Average with orders p and q) model is an explicit formula for X_t in terms of noise of the form

$$X_t = \varepsilon_t + \beta_1\varepsilon_{t-1} + \ldots + \beta_q\varepsilon_{t-q}.$$

The process is given by a (weighted) average of the noise, but not an average from time zero to the present time t; instead, an average moving with t is taken, using only the last $q + 1$ times. Using time lags we can write

$$X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right) \varepsilon_t.$$

Definition 7.1.6 (ARMA Model)**.** An ARMA(p, q) (AutoRegressive Moving Average with orders p and q) model is a discrete time linear equations with noise, of the form

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right) X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right) \varepsilon_t$$

or explicitly

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \ldots + \beta_q \varepsilon_{t-q}.$$

We may incorporate a non-zero average in this model. If we want that X_t has average μ, the natural procedure is to have a zero-average solution Z_t of

$$Z_t = \alpha_1 Z_{t-1} + \ldots + \alpha_p Z_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \ldots + \beta_q \varepsilon_{t-q}$$

and take $X_t = Z_t + \mu$, hence solution of

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \ldots + \beta_q \varepsilon_{t-q} + \widetilde{\mu}$$

with

$$\widetilde{\mu} = \mu - \alpha_1 \mu - \ldots - \alpha_p \mu.$$

Definition 7.1.7 (Difference operator - Integration)**.** The first difference operator, Δ, is defined as

$$\Delta X_t = X_t - X_{t-1} = (1 - L) X_t.$$

If we call

$$Y_t = (1 - L) X_t$$

then we may reconstruct X_t from Y_t by integration:

$$X_t = Y_t + X_{t-1} = Y_t + Y_{t-1} + X_{t-2} = \ldots = Y_t + \ldots + Y_1 + X_0$$

having the initial condition X_0. The second difference operator, Δ^2, is defined as

$$\Delta^2 X_t = (1 - L)^2 X_t$$

Assume we have

$$Y_t = (1 - L)^2 X_t$$

Then

$$\begin{aligned} Y_t &= (1 - L) Z_t \\ Z_t &= (1 - L) X_t \end{aligned}$$

so we may first reconstruct Z_t from Y_t :

$$Z_t = Y_t + \ldots + Y_2 + Z_1$$

where

$$Z_1 = (1-L)X_1 = X_1 - X_0$$

(thus we need X_1 and X_0); then we reconstruct X_t from Z_t :

$$X_t = Z_t + \ldots + Z_1 + X_0.$$

All this can be generalized to Δ^d, for any positive integer d.

Definition 7.1.8 (ARIMA models)**.** An ARIMA(p,d,q) (AutoRegressive Integrated Moving Average with orders p,d,q) model is a discrete time linear equations with noise, of the form

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)(1-L)^d X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t. \tag{66}$$

It is a particular case of ARMA models, but with a special structure. Set $Y_t := (1-L)^d X_t$. Then Y_t is an ARMA(p,q) model

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right) Y_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t$$

and X_t is obtained from Y_t by d successive integrations. The number d is thus the order of integration.

Example 7.1.9. The random walk is ARIMA $(0,1,0)$

We may incorporate a non-zero average in the auxiliary process Y_t and consider the equation

$$\left(1 - \sum_{k=1}^{p} \alpha_k L^k\right)(1-L)^d X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right)\varepsilon_t + \widetilde{\mu}$$

with

$$\widetilde{\mu} = \mu - \alpha_1\mu - \ldots - \alpha_p\mu.$$

7.1.2 Stationarity, ARMA and ARIMA processes

Under suitable conditions on the parameters, there are stationary solutions to ARMA models, called *ARMA* processes.

In the simplest case of AR(1) models, we have proved the stationarity (with suitable variance of the initial condition) when the parameter α satisfies $|\alpha| < 1$. In general, there are conditions but they are quite technical and we address the interested reader to the specialized literature. In the sequel we shall always use sentences of the form: "consider a stationary solution of the following ARMA model", meaning implicitly that it exists, namely that we are in the framework of such conditions. Our statements will therefore hold only in such case, otherwise are just empty statements.

Integration brakes stationarity. Solutions to ARIMA models are always non-stationary if we take Y_t stationary (in this case the corresponding X_t is called *ARIMA* process). For instance, the random walk is not stationary. The kind of growth of such processes is not always trivial. But if we include a non-zero average, namely we consider the case

$$\left(1-\sum_{k=1}^{p}\alpha_k L^k\right)(1-L)^d X_t = \left(1+\sum_{k=1}^{q}\beta_k L^k\right)\varepsilon_t + \widetilde{\mu}$$

then we have the following: if $d = 1, X_t$ has a linear trend; if $d = 2$, a quadratic trend, and so on. Indeed, at a very intuitive level, if Y_t is a stationary solution of the associated ARMA model, with mean μ, then its integration produces a trend: a single step integration gives us

$$X_t = Y_t + \ldots + Y_1 + X_0$$

so the stationary values of Y accumulate linearly; a two step integration produces a quadratic accumulation, and so on. When $\mu = 0$, the sum $Y_t + \ldots + Y_1$ has a lot of cancellations, so the trend is sublinear (roughly it behaves as a square root). But the cancellations become statistically not significant when $\mu \neq 0$. If $\mu > 0$ and $d = 1$ we observe an average linear growth; if If $\mu < 0$ and $d = 1$ we observe an average linear decay. This is also related to the ergodic theorem: since Y_t is stationary and its autocorrelation decays at infinity, we may apply the ergodic theorem and have that

$$\frac{Y_t + \ldots + Y_1}{t} \to E\left[Y_1\right] = \mu$$

(in mean square). Hence

$$Y_t + \ldots + Y_1 \sim \mu \cdot t.$$

There are fluctuations, roughly of the order of a square root, around this linear trend.

Bibliografia

[1] B. Scarpa A. Azzalini. *Data Analysis and Data Mining.* Oxford University Press, 2012.

[2] M. Barsanti F. Flandoli. *Corso di Probabilità e Processi Stocastici.* Scuola Normale Superiore di Pisa, 2010-11. URL: `http://users.dma.unipi.it/~flandoli/Automazione.html`.

[3] R. J. Hyndman G. Athanasopoulos. *Forecasting: Principles and Practice.* Third. Otexts, 2021.

[4] M. Guidolin. *Innovation Diffusion Processes: Concepts, Models and Predictions.* Department of Statistical Science.

[5] M. Guidolin. *Slides of the course "Business, Economics and Financial Data.*

[6] A. J. Izenman. *Modern Multivariate Statistical Techniques.* Springer, 2008.

[7] R. W. Keener. *Theoretical Statistics.* Springer, 2010.

[8] M. Guidolin R. Guseo. *Market potential dynamics in innovation diffusion: modelling the synergy between two driving forces.* Vol. 10. Working Paper Series. Department of Statistical Sciences, 2009.

[9] J. Friedman T. Hastie R. Tibshirani. *The Elements of Statistical Learning.* Second. Springer Series in Statistics. Springer, 2013.

[10] Wikipedia.

[11] G. Žitković. *Introduction to Stochastic Processes - Lecture Notes.* The University of Texas at Austin, 2010. URL: `https://web.ma.utexas.edu/users/gordanz/notes/introduction_to_stochastic_processes.pdf`.