Business Economic and Financial Data

Prof. *Don't care*
University of Padova

Written by: *Alessandro Manente*

Edited by: *Gianmarco Cracco*

Academic Year 2020/21

Compiled on January 13, 2021

Contents

Chapter 1

Forecasting

1.1 Introduction

Forecasting is a common statistical task in business, where it helps to inform decisions about the scheduling of production, transportation and personnel, and provides a guide to long-term strategic planning. However, business forecasting is often done poorly, and is frequently confused with planning and goals. They are three different things:

- **Forecasting**: is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts.
- **Goals**: are what you would like to have happen. Goals should be linked to forecasts and plans, but this does not always occur. Too often, goals are set without any plan for how to achieve them, and no forecasts for whether they are realistic.
- **Planning**: is a response to forecasts and goals. Planning involves determining the appropriate actions that are required to make your forecasts match your goals.

Forecasting should be an integral part of the decision-making activities of management, as it can play an important role in many areas of a company. Modern organisations require short-term, medium-term and long-term forecasts, depending on the specific application.

- **Short-term forecasts**: are needed for the scheduling of personnel, production and transportation. As part of the scheduling process, forecasts of demand are often also required.
- **Medium-term forecasts**: are needed to determine future resource requirements, in order to purchase raw materials, hire personnel, or buy machinery and equipment.
- **Long-term forecasts**: are used in strategic planning. Such decisions must take account of market opportunities, environmental factors and internal resources.

In the early stages of a forecasting project, decisions need to be made about what should be forecast. For example, if forecasts are required for items in a manufacturing environment, it is necessary to ask whether forecasts are needed for:

- every product line, or for groups of products?
- every sales outlet, or for outlets grouped by region, or only for total sales?
- weekly data, monthly data or annual data?

It is also necessary to consider the *forecasting horizon*. Will forecasts be required for one month in advance, for 6 months, or for ten years? Different types of models will be necessary, depending on what forecast horizon is most important.

How frequently are forecasts required? Forecasts that need to be produced frequently are better done using an automated system than with methods that require careful manual work.
It is worth spending time talking to the people who will use the forecasts to ensure that you understand their needs, and how the forecasts are to be used, before embarking on extensive work in producing the forecasts.
The appropriate forecasting methods depend largely on what data are available. If there are no data available, or if the data available are not relevant to the forecasts, then qualitative forecasting methods must be used. These methods are not purely guesswork: there are well-developed structured approaches to obtaining good forecasts without using historical data. **Quantitative forecasting** can be applied when two conditions are satisfied:

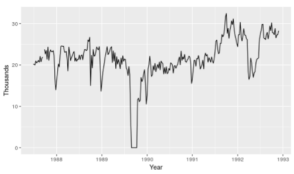
- numerical information about the past is available;
- it is reasonable to assume that some aspects of the past patterns will continue into the future.

Phases of forecasting process:

1. **Problem definition**: Often this is the most difficult part of forecasting. Defining the problem carefully requires an understanding of the way the forecasts will be used, who requires the forecasts, and how the forecasting function fits within the organisation requiring the forecasts. A forecaster needs to spend time talking to everyone who will be involved in collecting data, maintaining databases, and using the forecasts for future planning.
2. **Gathering information**: There are always at least two kinds of information required: (a) statistical data, and (b) the accumulated expertise of the people who collect the data and use the forecasts. Often, it will be difficult to obtain enough historical data to be able to fit a good statistical model. In that case, the judgmental forecasting methods can be used. Occasionally, old data will be less useful due to structural changes in the system being forecast; then we may choose to use only the most recent data.
3. **Preliminary (exploratory) analysis**: Always start by graphing the data. Are there consistent patterns? Is there a significant trend? Is seasonality important? Is there evidence of the presence of business cycles? Are there any outliers in the data that need to be explained by those with expert knowledge? How strong are the relationships among the variables available for analysis?
4. **Choosing and fitting models**: The best model to use depends on the availability of historical data, the strength of relationships between the forecast variable and any explanatory variables, and the way in which the forecasts are to be used. It is common to compare two or three potential models.
5. **Using and evaluating a forecasting model**: Once a model has been selected and its parameters estimated, the model is used to make forecasts. The performance of the model can only be properly evaluated after the data for the forecast period have become available.

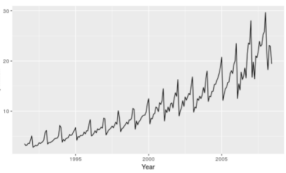
However, remember that good statistical models will handle evolutionary changes in the system; don't throw away good data unnecessarily!

Time series analysis: useful graphs



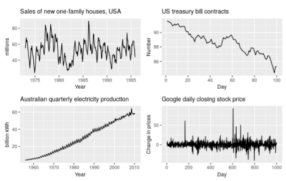
Weekly economy passenger load on an Airline Company

Time series analysis: useful graphs



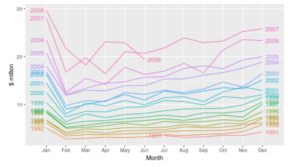
Monthly sales of antidiabetic drugs

Time series analysis: useful graphs



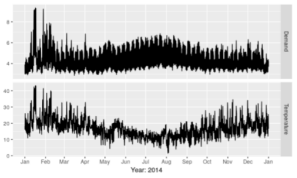
4 time series with different patterns

Time series analysis: useful graphs



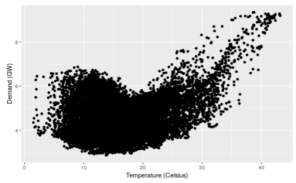
Monthly sales of antidiabetic drugs: 'seasonal plot'

Time series analysis: useful graphs



Electricity demand and temperatures in Australia (year 2014): relationship between series

Time series analysis: useful graphs



Electricity demand and temperatures in Australia (year 2014): relationship between series

1.2 Autocorrelation

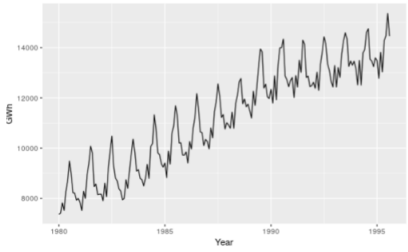
Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between **lagged values of a time series**.

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

t=1 ??

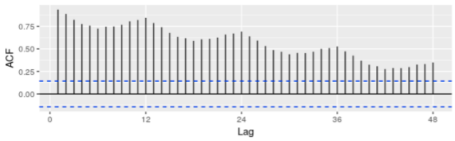
The autocorrelation coefficients are plotted to show the **autocorrelation function** or ACF. The plot is also known as a **correlogram**.

Autocorrelation



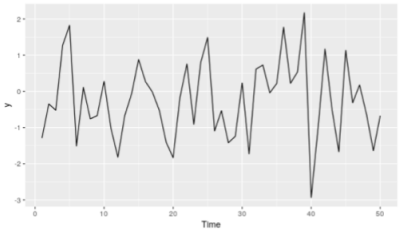
Monthly electricity demand in the period 1980-1995

Autocorrelation



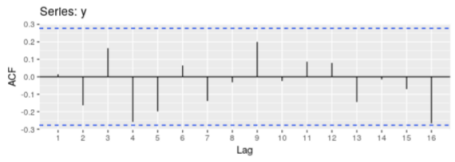
Monthly electricity demand in the period 1980-1995: correlogram

Autocorrelation



Time series that show no autocorrelation are called *White Noise* process

Autocorrelation



White Noise process: autocorrelation

Chapter 2

Multiple Linear Regression

Let us recall the **multiple linear regression** model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

where X_j is 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

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

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

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

We make the following assumptions regarding error terms $(\varepsilon_1, \ldots, \varepsilon_N)$

1. errors have mean zero
2. errors are uncorrelated
3. errors are uncorrelated with $X_{j,i}$

The R^2 statistic is given by

$$R^2 = \frac{\sum \left(\hat{Y}_i - \bar{Y}\right)^2}{\sum \left(Y_i - \bar{Y}\right)^2} = \frac{\mathrm{ESS}}{\mathrm{TSS}}$$

In addition to looking at the R^2, it can be useful to plot the data. Graphical summaries may reveal problems with a model that are not visible from numerical statistics.
In order to test the global significance of the model we

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_k = 0$$
$$H_1 : \text{ at least one } \beta_j \neq 0$$

through the F statistic

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

Results may be usefully displayed in an ANOVA table

Source	df	SS	MS	F
Model	p	ESS	MSR	MSR/MSE
Error	n − p − 1	RSS	MSE	
Total	n − 1	SST		

After examining the global significance of the model, it is useful to evaluate the significance of parameters. The hypothesis system is

$$H_0 : \beta_j = 0$$
$$H_1 : \beta_j \neq 0$$

and the test is defined as

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

where b_j is the estimate of the j_{th} coefficient and $\text{se}(b_j)$ is the standard error.

2.0.1 Example

Let us consider a sample of 10 stores. We want to measure the effect exerted on weekly sales Y from 3 predictors: X_1 : size $\left(\text{ in } m^2\right)$ X_2 : expenditures in promotional activities X_3 : population density $\left(\text{per } km^2\right)$

We estimate the model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$

coefficient	estimate	std. error	t-statistic
β_0	-22.90	16.7729	-1.36
β_1	0.70	0.1949	3.58
β_2	0.52	0.1709	3.02
β_3	-0.36	0.3822	-0.94

Source	df	SS	MS	F
Model	3	23348.9	7782.98	23.48
Error	6	1989.12	331.521	
Total	9	25338.1		

$R^2 = 0.921497$

Previous results suggest to remove X_3, and estimate a reduced model with variables X_1 and X_2.

coefficient	estimate	std. error	t-statistic
β_0	-21.34	16.5612	-1.28
β_1	0.54	0.0966	5.59
β_2	0.49	0.1687	2.96

Source	df	SS	MS	F
Model	2	23053.5	11526.7	35.32
Error	7	2284.59	326.369	
Total	9	25338.1		

$R^2 = 0.909836$

~~Previous results suggest to remove X_3, and estimate a reduced model with variables X_1 and X_2.~~

~~coefficient estimate std. error t-statistic~~
~~β_0 -21.34 16.5612 -1.28~~
~~β_1 0.54 0.0966 5.59~~
~~β_2 0.49 0.1687 2.96~~

~~Source df SS MS F~~
~~Model 2 23053.5 11526.7 35.32~~
~~Error 7 2284.59 326.369~~
~~Total 9 25338.1~~

~~$R^2 = 0.909836$~~

How do we interpret these results?

When we fit a linear regression model to a particular data set, many problems may occur. Most common among these are the following:

- Non-linearity of the response-predictor relationships
- Correlation of error terms
- Non-constant variance of error terms
- Outliers
- Collinearity

we will discuss some of these problems in more detail ...

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

- reduces the accuracy of estimates of the regression coefficients
- the standard error for β_j grows
- the t-statistic declines $\rightarrow$ we may fail to reject $H_0 : \beta_j = 0$

how do we detect a problem of collinearity?

- a simple way to detect collinearity is to look 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 $\rightarrow$ collinearity
- it is possible for collinearity to exist between three or more variables $\rightarrow$ **multicollinearity**

A better way to assess the multicollinearity is to compute the variance inflation factor, VIF.

$$\text{VIF} = \frac{1}{1 - R_j^2}$$

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 VIF $_j > 1$. For example, $R_j^2 = 0.9$, VIF $_j = 10$.

2.0.2 Example

Let us consider a sample of 10 households and the following variables:

- Y : yearly amount spent in food (hundreds eur)
- X_1 : family income (thousands eur)
- X_2 : number of family members

We first calculate the correlation matrix ...

	Y	X_1	X_2
Y	1	0.884	0.737
X_1		1	0.867
X_2			1

We estimate the model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$

coefficient	estimate	std. error	t-statistic
β_0	3.51865	3.16055	1.1133
β_1	2.27762	0.81261	2.80284
β_2	-0.411406	1.23603	-0.332844

Source	df	SS	MS	F
Model	2	213.422	106.711	12.75
Error	7	58.578	8.3682	
Total	9	272		

$R^2 = 0.7846$

We estimate the model $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$

coefficient	estimate	std. error	t-statistic
β_0	3.51865	3.16055	1.1133
β_1	2.27762	0.81261	2.80284
β_2	-0.411406	1.23603	-0.332844

Source	df	SS	MS	F
Model	2	213.422	106.711	12.75
Error	7	58.578	8.3682	
Total	9	272		

R^2 = 0.7846 How do we interpret these results?

Let us compute the Variance Inflation Factor. This may be easily computed for X_1 e X_2 considering that

$$R^2 = (r_{X_1X_2})^2 = (0.867)^2 = 0.75 \text{ so that}$$

$$\text{VIF}_{X_1} = 1/(1 - 0.75) = 4$$

$$\text{VIF}_{X_2} = 1/(1 - 0.75) = 4$$

There is a multicollinearity problem: solution $\rightarrow$ remove X_2 from the model and estimate a simple regression with X_1.

Chapter 3

Multiple linear regression with 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 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

$$Y_t = \beta_0 + \beta_1 t + \beta_2 S_2 + \beta_3 S_3 + \beta_4 S_4 + \varepsilon_t$$

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.
Time series data tend to be **autocorrelated**. Autocorrelation occurs when the over time. For example, a change in p on both current and future sales. Autocorrelation may be detected throuhgh a *graphical inspection of residuals.* Specific tests on residuals.
A typical example of autocorrelation is defined as

$$Y_t = \beta_o + \beta_1 X_t + \varepsilon_t$$

with

$$\varepsilon_t = \rho \varepsilon_{t-1} + \nu_t$$

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

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

The Durbin-Watson test is defined as

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

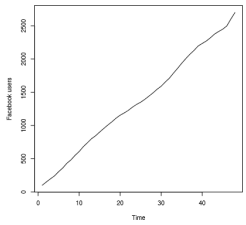
The values of DW range between 0 and 4 with a central value of 2 . For large samples, the following holds

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

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?

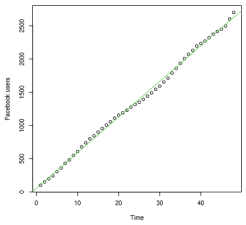
Example

Facebook users: quarterly data 2008-2020



Example

Facebook users: simple linear regression



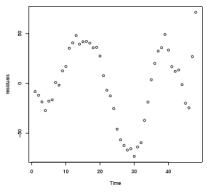
Example

Facebook users: simple linear regression

```
lm(formula = fb ~ time)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  54.5363    10.9917   4.962    1e-05 ***
time          53.6507     0.3905 137.378   <2e-16 ***
---
Residual standard error: 37.48 on 46 degrees of freedom
Multiple R-squared:  0.9976,Adjusted R-squared:  0.9975
F-statistic: 1.887e+04 on 1 and 46 DF,  p-value: < 2.2e-16
```

Example

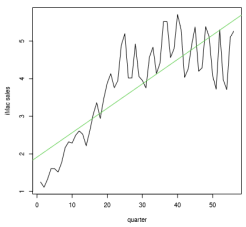
Facebook users: residuals



Durbin-Watson test: DW = 0.16378, p-value < 2.2e-16
Positive autocorrelation in residuals

Example

iMac sales: simple linear regression



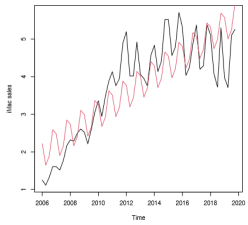
Example

iMac sales: linear regression with trend and seasonality

```
Call:
tslm(formula = mac.ts ~ trend + season)

Residuals:
     Min       1Q   Median       3Q      Max
-1.60158 -0.42293 -0.00687  0.54972  1.42797

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.155255   0.236078   9.129 2.62e-12 ***
trend        0.064591   0.005613  11.507 8.68e-16 ***
season2     -0.640448   0.256052  -2.501   0.0156 *
season3     -0.460039   0.256237  -1.795   0.0785 .
season4      0.176727   0.256544   0.689   0.4940
---

Residual standard error: 0.6773 on 51 degrees of freedom
Multiple R-squared:  0.7436,Adjusted R-squared:  0.7235
F-statistic: 36.97 on 4 and 51 DF,  p-value: 1.695e-14
```

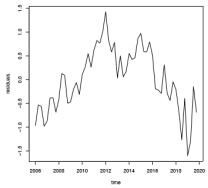
Example

iMac sales: linear regression with trend and seasonality



Example

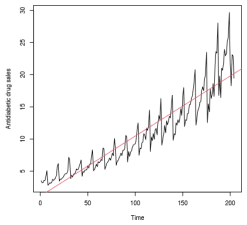
iMac sales: residuals



Residuals clearly show a nonlinear behaviour

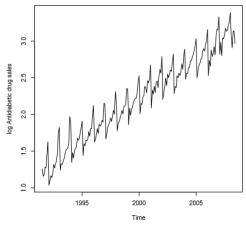
Example

Monthly sales of a drug: simple linear regression



Example

Monthly sales of a drug: log transformation



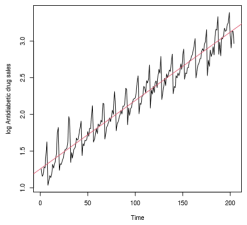
Example

Monthly sales of a drug: simple linear regression with log transformation

```
Call:
lm(formula = la10 ~ t)

Residuals:
     Min       1Q   Median       3Q      Max
-0.36954 -0.09621 -0.00889  0.07139  0.43395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.2577135  0.0216920   57.98   <2e-16 ***
t           0.0093211  0.0001835   50.80   <2e-16 ***
---

Residual standard error: 0.1543 on 202 degrees of freedom
Multiple R-squared:  0.9274,Adjusted R-squared:  0.927
F-statistic:  2580 on 1 and 202 DF,  p-value: < 2.2e-16
```

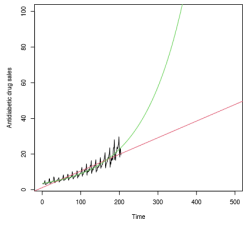
Example

Monthly sales of a drug: log transformation



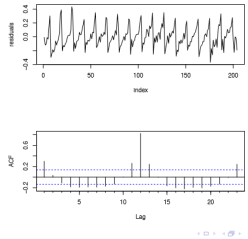
Example

Monthly sales of a drug: model comparison



Example

Monthly sales of a drug: residuals



Chapter 4

Nonlinear Model for New Product Growth

New product life cycle phases:

1. Introduction
2. Growth
3. Maturity
4. Decline

What are the variables influencing a product's life cycle? Marketing strategies play an essential role ... but the success of a new product ultimately depends on consumers accepting them.
Diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system (Rogers, 2003). Four key elements for describing an innovation diffusion process: innovation, communication channels, time, social system.
An innovation is:

- New product, new service, new technology, new production process, new way of doing things (Schumpeter, 1947).
- Typical distinction: radical vs incremental innovations.
- Radical innovations could be hindered from barriers and social inertia.

General aim: depict the successive increases in the number of adopters and predict the continued development of a diffusion process already in progress (Mahajan and Muller, 1979).

- Fourt and Woodlock model (1960)
- Mansfield model (1961)
- Bass model (1969)
- Generalized Bass model (1994)

4.1 Bass Model

The Bass Model is defined by a **first order differential equation**

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

where: p = innovation, q = imitation, $q\frac{z(t)}{m}$ = word of mouth.

If we pose $\frac{z(t)}{m} = y(t)$ the model becomes

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

The Bass Model has a closed-form solution

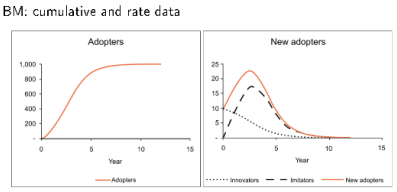
$$y(t) = F(t; p, q) = \frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}} \quad t > 0$$

or, by posing $z = ym$

$$z(t) = mF(t; p, q) = m\frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}} \quad t > 0$$

Cumulative sales $z(t)$ 'depend' on parameters p and q. The market potential m is a scale parameter and is assumed *constant*.

Bass Model



4.1.1 Estimation

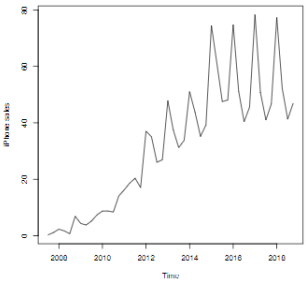
The Bass Model is a nonlinear model

$$Z(t) = f(\beta, t) + \varepsilon(t)$$

where $Z(t)$ is the dependent variable, $f(\beta, t)$ is the deterministic term, function of $\beta \in \mathbb{R}^k$ and of time t. The second term, $\varepsilon(t)$, is the error term, for which usual assumptions hold, namely $E(\varepsilon(t)) = 0, \text{Var}(\varepsilon(t)) = \sigma^2$, $\text{Cov}(\varepsilon(t), \varepsilon(t')) = 0, t \neq t'$

Typical starting values for p and q are 0.01 and 0.1. Estimating m is the most difficult task. Parameter estimates are very sensitive to the number of available data. Reliable estimates are obtained after the maximum peak, but "By the time sufficient observations have been developed for realiable estimation, it is too late to use the estimates for forecasting purposes" (Mahajan, Muller, Bass, 1990).

Example: Apple iPhone life cycle



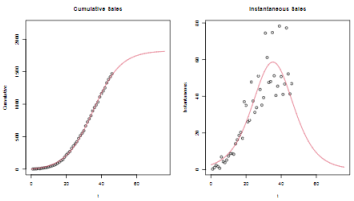
Cumulative quarterly sales data from 2007 to 2019 (source: Apple Inc.)

Example: Apple iPhone life cycle

Bass Model for iPhone life cycle: parameter estimates and 95% CIs

```
Estimate
       Estimate    Std.Error        Lower        Upper  p-value
m : 1.823747e+03 3.412507e+01 1.756863e+03 1.890631e+03 5.84e-41
p : 1.412817e-03 5.410927e-05 1.306765e-03 1.518869e-03 5.26e-28
q : 1.258732e-01 2.675751e-03 1.206289e-01 1.311176e-01 1.29e-38

R-squared
0.9995498
```

Example: Apple iPhone life cycle



Cumulative and instantaneous sales data and forecasts with BM

It is a **parsimonious model** with just three parameters m, p, q. Only needs aggregate sales data. Easy to interpret.

Its limitations are:

- The market potential m is constant along the whole life cycle.
- The Bass Model does not account for marketing mix strategies.
- It is a model for products with a limited life cycle: needs a hypothesis.

4.2 Generalized Bass Model

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.
The Generalized Bass Model (Bass et al., 1994) adds an *intervention function* $x(t)$

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

where $x(t)$ is an integrable, non negative function. The Bass Model is a special case where $x(t) = 1$.
If $0 < x(t) < 1$ the process *slows down*, if $x(t) > 1$ the *process accelerates*.
The closed-form solution of the model is

$$z(t) = m\frac{1 - e^{-(p+q)\int_0^t x(\tau)d\tau}}{1 + \frac{q}{p}e^{-(p+q)\int_0^t x(\tau)d\tau}}, \quad t > 0$$

Interesting: function $x(t)$ does not modify the market potential m! Function $x(t)$ modifies the speed of the process.
Function $x(t)$ may take several forms in order to describe various types of shock. A strong and fast shock may take an exponential form

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

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

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

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.
~~EXAMPLES.~~
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,

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

Other combinations are possible.
The usual performance indicator is the R^2

$$R^2 = \frac{\text{SST} - \text{SSE}}{\text{SST}} = \frac{\sum (y_i - \bar{y})^2 - \sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

where $y_i, i = 1, 2, \ldots, n$ are calculated with the selected model. Further evaluations are performed through analysis of residuals (e.g. residual plots, Durbin-Watson statistic).
In order to select between two 'nested' models, a suitable tool is the $\tilde{R}^2$

$$\tilde{R}^2 = \frac{\text{SSE}_{m_1} - \text{SSE}_{m_2}}{\text{SSE}_{m_1}} = \left(R^2_{m_2} - R^2_{m_1}\right) / \left(1 - R^2_{m_1}\right)$$

where $R^2_{m_i}, i = 1, 2$ is the R^2 of model m_i. If $\tilde{R}^2 > 0.3$ then the more complex model is significant.
A generalization of the Bass Model considers a dynamic market potential, $m(t)$

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

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

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.
Market of new products is unstable and uncertain in the first phase of diffusion: 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
We may notice that $m(t)$ is 'free'

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

The GGM (Guseo and Guidolin, 2009) is a generalization of the Bass Model, where $m(t)$ is time-dependent

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

and function of a communication process

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

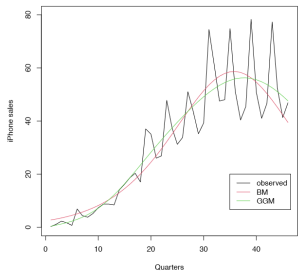
Example: Apple iPhone

GGM for iPhone: estimates and 95% CIs

	Estimate	Std.Error	Lower	Upper	P-value
K	2116.78	97.50	1925.68	2307.88	< 0.0001
p_c	0.00059	0.00	0.0028	0.009	< 0.0001
q_c	0.21	0.04	0.13	0.28	< 0.0001
p_s	0.0021	0.00	0.0015	0.0026	< 0.0001
q_s	0.10	0.01	0.09	0.11	< 0.0001

$R^2 = 0.99986$

Example: Apple iPhone



Model comparison ...

Chapter 5

Time Series Analysis

5.1 Forecasting Accuracy

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

$$\text{ME} = \frac{1}{n}\sum_{t=1}^{n} e_t$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |e_t|$$

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$$

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.

$$\text{PE}_t = \frac{Y_t - F_t}{Y_t} * 100$$

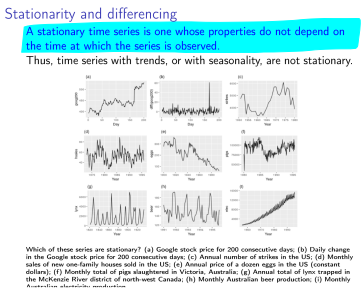
$$\text{MPE} = \frac{1}{n}\sum_{t=1}^{n} PE_t$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} |PE_t|$$

5.2 ARIMA Models

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.*

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.



5.2.1 Differencing

Literally taking the difference

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.

$$y'_t = y_t - y_{t-1}$$

Seasonal differencing (for monthly data)

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

A further differencing may be performed

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

5.2.2 Backshift Notation

The backward shift operator B is a useful notational device when working with time series lags:

$$By_t = y_{t-1}$$

In other words, B has the effect of shifting the data back one period. Two applications of B shifts the data back two periods

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

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

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

5.2.3 Autoregressive Models

In a multiple regression model, we forecast the variable of interest using a linear combination of predictors.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

In an autoregression model, we forecast the variable of interest using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself.

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

This is like a multiple regression but with lagged values of y_t as predictors. We refer to this as an $AR(p)$, an autoregressive model of order p.

5.2.4 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.

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

We refer to this as an $MA(q)$, a Moving Average of order q.

5.2.5 ARIMA 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. 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

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

or, by using backshift notation,

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

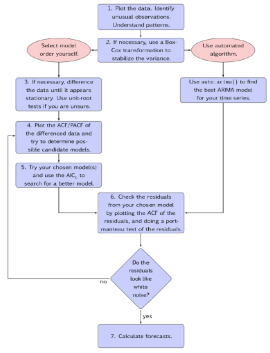
If an ARMA (p, q) model is non stationary, we obtain an ARIMA (p, d, q) model. The simplest case, ARIMA $(1, 1, 1)$, is defined as

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

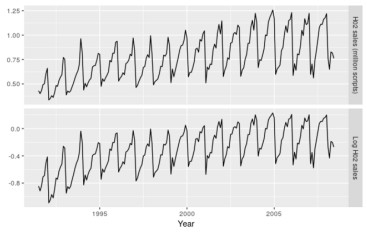
The general form of an ARIMA (p, d, q) may produce a great variety of ACF and PACF.
A further extension to ARMA models concerns seasonality. An ARIMA model with seasonal components is an ARIMA $(p, d, q)(P, D, Q)_s$, where (p, d, q) indicates the non-seasonal part of the model, while (P, D, Q) indicates the seasonal part of order s. The ARIMA model (1,1,1)$(1, 1, 1)_4$ is

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

Model selection

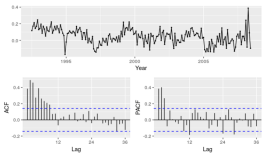


Example



A drug's sales (July 1991- June 2008) What are the main features of the series?

Example



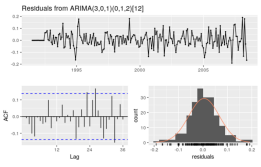
Seasonal differencing

Example

Model	AICc
ARIMA(3,0,1)(0,1,2)$_{12}$	-485.5
ARIMA(3,0,3)(1,1,1)$_{12}$	-484.2
ARIMA(3,0,1)(0,1,1)$_{12}$	-483.7
ARIMA(3,0,3)(1,1,0)$_{12}$	-476.3
ARIMA(3,0,0)(2,1,0)$_{12}$	-475.1
ARIMA(3,0,2)(2,1,0)$_{12}$	-474.9
ARIMA(3,0,1)(1,1,0)$_{12}$	-463.4

Different models have been estimated and compared on the basis of the AIC

Example



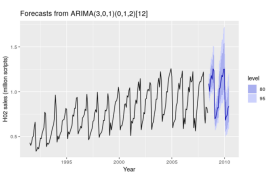
Are residuals white noise?

Example

Model	RMSE
ARIMA(3,0,1)(0,1,2)$_{12}$	0.0622
ARIMA(3,0,3)(1,1,1)$_{12}$	0.0630
ARIMA(2,1,4)(0,1,1)$_{12}$	0.0632
ARIMA(2,1,3)(0,1,1)$_{12}$	0.0634
ARIMA(3,0,3)(0,1,1)$_{12}$	0.0639
ARIMA(2,1,5)(0,1,1)$_{12}$	0.0640
ARIMA(3,0,1)(0,1,1)$_{12}$	0.0644
ARIMA(3,0,2)(0,1,1)$_{12}$	0.0644
ARIMA(3,0,2)(2,1,0)$_{12}$	0.0645
ARIMA(3,0,3)(2,1,0)$_{12}$	0.0646
ARIMA(4,0,2)(0,1,1)$_{12}$	0.0648
ARIMA(4,0,3)(0,1,1)$_{12}$	0.0648
ARIMA(3,0,0)(2,1,0)$_{12}$	0.0661
ARIMA(3,0,1)(1,1,0)$_{12}$	0.0679

Test set (july 2006–june 2008)
Auto.arima and model comparison by RMSE

Example



Forecast with the selected model

5.3 Short-Term Forecasting: Simple Exponential Smoothing

The simple exponential smoothing is defined as

$$F_{t+1} = F_t + \alpha\,(Y_t - F_t)$$

where α is a constant term taking values between 0 e 1. The new forecast F_{t+1} is the old forecast F_t with an adjustment.
An equivalent way to express the simple exponential smoothing is

$$F_{t+1} = \alpha Y_t + (1-\alpha)F_t$$

The new forecast F_{t+1} is a weighted average of the last observation, Y_t, and the last forecast, F_t
Why 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

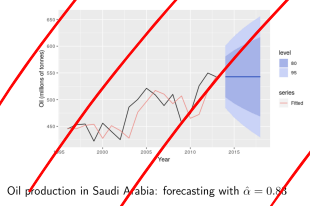
$$\begin{aligned} F_{t+1} =& \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} \\ &+ \alpha(1-\alpha)^3 Y_{t-3} + \ldots + \alpha(1-\alpha)^{t-1}Y_1 + (1-\alpha)^t F_1 \end{aligned}$$

Initialization of the process

$$F_2 = \alpha Y_1 + (1-\alpha)F_1$$

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. A higher value for α is more sensitive to a change in the data structure, while a lower value generates a 'flat' forecast. A suitable selection for α is that minimizing the MSE.

Example



Oil production in Saudi Arabia: forecasting with $\hat{\alpha} = 0.83$

5.4 Short-Term Forecasting: Holt's Exponential Smoothing

The Holt's linear trend method is a useful extension 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)

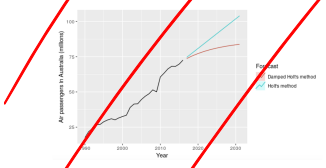
$$\begin{aligned} L_t &= \alpha Y_t + (1-\alpha)\,(L_{t-1} + b_{t-1}) \\ b_t &= \beta\,(L_t - L_{t-1}) + (1-\beta)b_{t-1} \\ F_{t+m} &= L_t + b_t m \end{aligned}$$

L_t denotes an estimate of the level of the series at time t and b_t an estimate of the slope t. This exponential smoothing is a double smoothing. The forecast function is no longer flat but trending. 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.
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. A useful extension includes a damping parameter $0 < \phi < 1$

$$L_t = \alpha Y_t + (1-\alpha)\left(L_{t-1} + \phi b_{t-1}\right)$$
$$b_t = \beta\left(L_t - L_{t-1}\right) + (1-\beta)\phi b_{t-1}$$
$$F_{t+m} = L_t + \left(\phi + \phi^2 + \cdots + \phi^h\right) b_t$$

Example

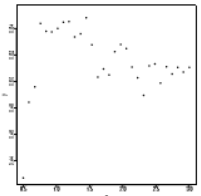


Airline passengers:
Holt's and Dampened Holt's exponential smoothing $\phi = 0.90$

Chapter 6

Bias/Variance Trade-Off

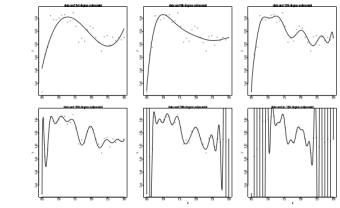
A simple prototype problem



Yesterday we observed n couples (x_i, y_i), for $i = 1, \ldots, n$, of data ($n = 30$). These data are artificially generated by the law $y = f(x) +$ error where $f(x)$ is a unspecified smooth and regular function. We wish to obtain a rule (model), like $\hat{y} = \hat{f}(x)$, that enables us to **predict** y once we know x; a rule that allows us to predict y as new observations of x become available, say tomorrow.

A simple possibility is to interpolate data with a polynomial but, of which degree? $0, 1, 2, \ldots, 29$? Let's try to use polynomials of degree p (with $p = 0, 1, \ldots, n-1 = 29$). We need to estimate p parameters $\left(+\sigma^2\right)$.

A simple prototype problem

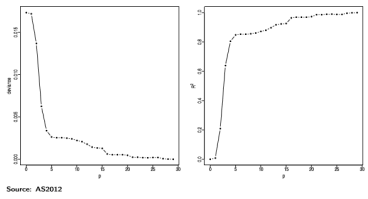


Source: AS2012

By growing of p the fitting of the polynomials is getting better.

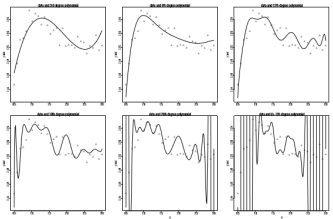
A simple prototype problem

We measure the goodness of fit by obtaining, for each p the residual deviance and the coefficient of determination R^2.



Source: AS2012

A simple prototype problem



Source: AS2012

Tomorrow we will receive a new set of n data $\{y_i, i = 1, \ldots, n\}$, generated by the same phenomenon of the yesterday data, that is, the same function $f(x)$

We want to predict these new observations, by assuming (for simplicity) that the new y_i are associated to the same x_i of the yesterday data.

We compare our predictions (one for each polynomial) with the new data observed tomorrow.

EXAMPLES.

If we knew $f(x)$ . . . We want to estimate $f(x)$ using a generic estimator $\hat{y} = \hat{f}(x)$ (in our example, can be one of the 30 fitted polynomials) We start by considering a specific value x' of x, among the n observed.

If we knew the mechanism used to generate the data precisely, we knew also $f(x')$, and we could calculate some quantities of interest to evaluate the estimator $\hat{y}$. For example, an important goodness-of-fit indicator is the mean squared error (with respect to the random variable y)

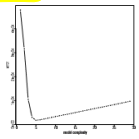
$$\mathbb{E}_y \left\{ \left[\hat{y} - f(x') \right]^2 \right\}$$

since we are not interested only on the single point x', we consider the sum of the mean squared errors

for all the n values of x,

$$\sum_{i=1}^{n} \mathbb{E}_y \left\{ [\hat{y} - f(x_i)]^2 \right\}$$

If we do it for all the possible choices of p, which is an indicator of the model complexity, we may obtain the plot



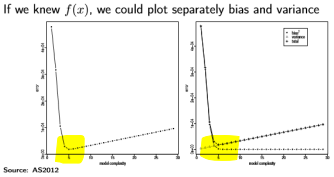
Even if the true $f(x)$ is not a polynomial, there exists a degree p which is better than the others

The mean squared error may be divided in two components

$$\begin{aligned} \mathbb{E}\left\{ [\hat{y} - f(x')]^2 \right\} &= \mathbb{E}\left\{ [\hat{y} \pm \mathbb{E}\{\hat{y}\} - f(x')]^2 \right\} \\ &= [\mathbb{E}\{\hat{y}\} - f(x')]^2 + \mathrm{var}\{\hat{y}\} \\ &= \text{bias}^2 + \text{variance} \end{aligned}$$

A trade-off

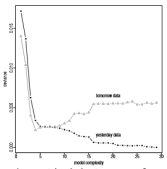
If we knew $f(x)$, we could plot separately bias and variance



Source: AS2012

A simple prototype problem

- But as we do not know $f(x)$, we only may compute the *residual variance* for the new (*tomorrow*) data:



This plot gives the residual deviance as function of the degree p, by using the model obtained with the *yesterday* data to predict the *tomorrow* data

When p (the model complexity indicator) increases, the fit improves on the yesterday data, but this is not true for the tomorrow data. Goodness-of-fit measure is not a good indicator of the quality of the model. When p increases too much, we '**overfit**' the data and this indicates an excess of optimism! This happens because the model (the polynomial in the example) follows random fluctuations in yesterday's data not observed in the new sample (and not characteristic of the studied phenomenon), and it mistakes local (random) regularity with a systematic pattern.
So that. .. do not evaluate a model by using the same data used to fit it (the yesterday ones). If we want a more reliable evaluation, we need to use other data (the tomorrow ones). How?!
We need tools in order to select models: training set and test (evaluation) set, cross-validation, information criteria.
If we have n data, and n is large, we can divide it in two groups randomly chosen: a **training set** used to fit the various candidate models and a **test set** (sometime called evaluation set) used to evaluate the performance of the available models and to choose the most accurate one. We compare results obtained with different models on the test set. This scheme reduces the sample size used for fitting the model, but this is not a problem when n is huge. Because the same test set can be used to evaluate many different models, there is a risk that the final assessment is still somewhat biased and too optimistic. Sometimes a third set of data, called **validation set**, is often created and used for final evaluation of the prediction error Examples of proportions for the sizes of the sets are:

training set	test set	validation set%
50%	25%	25%
75%	25%	–

training and test sets are somehow similar to what was done with yesterday and tomorrow data.
When n is not very large, we may modify the previous schema: presume that we use 75% of the data for training and 25% for testing the models (but the following schema is valid whatever portion we choose). for greater accuracy, we do not want to assign only that specific 25% of the data to the role of test set Also, if n is not very large and we only use 75% of the data to fit the model, the estimate

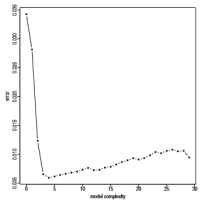
will be further impoverished (high variability), whereas we would like to take better advantage of available information.
One way of partially overcoming this arbitrariness is to split the data into four equal parts and then use three portions in rotation for training the model and the remaining portion for testing it.

We then cross the role of the data sets: one of the portions used as the training set is now used as a test set, and the 'old' test set is incorporated into the training set with the other two portions. Obviously, this scheme requires four iterations of the training and testing procedures. An *average* (or some other combination) of the four different predictions can be used as final prediction.
The procedure becomes progressively more accurate if, instead of 4 parts sized $n/4$, we use $k(k > 4)$ portions of size n/k and repeat the operations k times. and it is more effective when large values of k are used. Clearly, the computational burden of this procedure increases considerably as k increases, but when n is not too large, this is a good way to better exploit all the data.

6.0.1 Cross-Validation (Leave-One-Out)

The maximum possible value for k is n. To fit the model, $n-1$ observations are used, and the remaining observation is used for testing. Once we have rotated the only datum serving as test set, we must perform a total of n fitting operations this procedure is known as *leave-one-out cross-validation.*
There are theoretical results guaranteeing that, when $n \to \infty$, this procedure certainly leads us to select the most appropriate model.
Algorithm
(don't want to write it in a decent manner) 1. Read n records of x and y. 2. Cycle for $p = 0, 1, \ldots, \max_p$
(a) cycle for $i = 1, \ldots, n$: i) fit the model of degree p by eliminating the i th observation, ii) obtain prediction $\hat{y}_{-i}$ for y_i corresponding to point x_i, iii) obtain error $e_i \leftarrow (y_i - \hat{y}_{-i})$ (b) calculate $D^*(p) \leftarrow \sum_{i=1}^{n} e_i^2$ 3. Choose p so that $D^*(p)$ is minimum.

cross-validation – example



Source: AS2012
We choose p minimising $D^*(p)$ in the algorithm (the sum of the squared errors); in our example we obtain $p = 4$.

6.0.2 Information Criteria

The residual variance (or the deviance) is an unreliable indicator of the quality of the model, because it is too optimistic in evaluating the prediction error. We can penalize the deviance $D = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ or a monotonic transform: $-2 \log L = n \log(D/n) + (\text{ costant })$ with a suitable quantity quantifying the model complexity The $\log L$, for the gaussian model, has an interpretation as log-likelihood. Criteria that follow this logic can be traced back to objective functions such as

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

The choice of the specific penalty function identifies a particular criterion.
how to choose the penalty (p)? For the theory of likelihood we know that, when we compare nested models

$$(2 \log L_{p_1} - 2 \log L_{p_2}) \sim \chi^2_{p_1 - p_2}$$

(under H_0) When we add one parameter (degree from p to $p+1$), the average increment of $2\log L$ is 1, if the model does not need the extra-parameter.

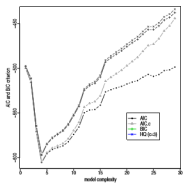
Following this reasoning, the penalization term will be at least p

Some possibile penalty are in the following table

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

These criteria are applied also to not nested and not gaussian models, by using the appropriate $\log L$

Information criteria – example



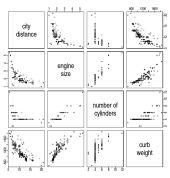
We choose p minimising $IC(p)$ using some criteria in the previous table;
in our example all choices for penalty suggests $p = 4$.

Chapter 7

Non Parametric Regression

- A variety of smoothing techniques
- Kernel methods and local regression
- Spline based methods
- Bandwidth selection
- Linear smoothers and equivalent kernels

Back to polynomial regression...



- we want identify a relationship allowing the prediction of the consumption of fuel (distance covered per unit of fuel) as a function of certain characteristics of a car.
- We consider data for $n = 203$ models of cars in circulation in 1985 in the United States but produced elsewhere.
- 27 variables are available, 4 of which are shown in figure

We start considering only one explanatory variable x = engine size we still refer to the generic formulation

$$y = f(x) + \epsilon$$

but since we don't believe that $f(x)$ is a polynomial, we make no reference to any parametric formulation for $f(x)$ without assuming that a specific parametric class of functions ... we try to leave data 'talk' *freely.*
In fact we still assume 'something' about the data, but with a weaker formulation in terms of assumptions. The approach may have several very different formulations. Thus, the "free" expression of the data is not in fact completely free: there are various methods available, and using one rather than another may produce different results, at least partially or in certain circumstances. The best tool for a specific problem is ultimately our choice.
EXAMPLE.
We are interested in penalising the prediction errors on the y by using a *quadratic loss function*

$$L(y, f(x)) = (y - f(x))^2$$

If we choose to measure errors by average squared error

$$\mathbb{E}\left\{(y - f(x))^2\right\}$$

The minimum solution is

$$f(x) = \mathbb{E}\{y \mid x = x_0\}$$

that is the **regression function**.
We want to estimate this regression function:

$$f(x) = \mathbb{E}\{y \mid x\}$$

through a model

$$y = f(x) + \varepsilon$$

ε : random error Data: n pairs $(x_i, y_i), i = 1, 2, \ldots, n$.
Goal: estimate f.
The regression function suggests a data implementation. To predict y at $x = x_0$, gather all the training pairs (x_i, y_j) having $x_i = x_0$, then to estimate $f(x_0)$, use the mean of their y_i :

$$\hat{f}(x_0) = \text{Average}(y \mid x = x_0)$$

Problem: in the training data, there may be no observations having

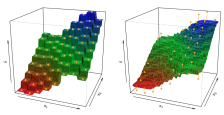
$$x_i = x_0$$

7.1 Nearest Neighbour Averaging

Estimate $\mathbb{E}\{y \mid x = x_0\}$ by averaging those y_i whose x_i are in a neighbourhood of x e.g. define the neighbourhood to be the set of k observations having values x_i closest to x_0 in euclidean distance $\|x_i - x_0\|$ (in the univariate case this is the absolute value $|x_i - x_0|$). This method is called **nearest neighbour**.
Given a value k and a prediction point x_0, the KNN regression identifies in the training set the k nearest observations, N_0

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

k-Nearest Neighbors: regression

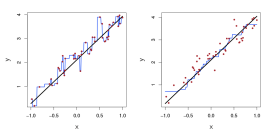


KNN with $p = 2$, $k = 1$ (left) and $k = 9$ (right). With small k high variance and low bias, since prediction is performed on a single observation.

The optimal value of k is related to the *trade-off viarance-bias.* small $k \rightarrow$ high variance and low bias big $k \rightarrow$ low variance (smoother prediction) and high bias local structure of $f(X)$ may not be captured-

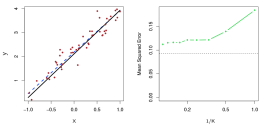
k-Nearest Neighbors: regression

Parametric approach may be preferred to the non parametric if the parametric form is close to the 'real' f.



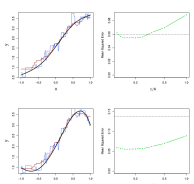
Comparison between KNN with $k = 1$ (left) e $k = 9$ (right). Since the true relationship is linear the non parametric approach will have a worse performance.

k-Nearest Neighbors: regression



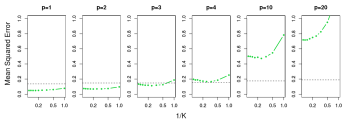
Regression line (dashed line) Test MSE for regression line (dashed) and KNN (green) as function of $1/k$.
Best results for KNN are with high value of k.

k-Nearest Neighbors: regression



Nonlinear relationships and KNN with $k = 1$ (blue) and $k = 9$ (red). Conditional to nonlinearity of f the KNN performance changes with respect to LM. As the nonlinearity becomes more evident, the performance of KNN with high k will increase.

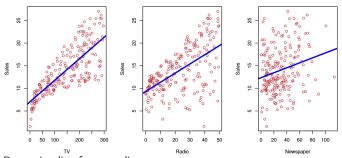
k-Nearest Neighbors: regression



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' . . . curse of dimensionality

k-Nearest Neighbors: example

Sales of a product in thousands of units as function of budget in `tv`, `radio`, `newspapers` for 200 different markets.



Regression line for `tv`, `radio`, `newspapers`.

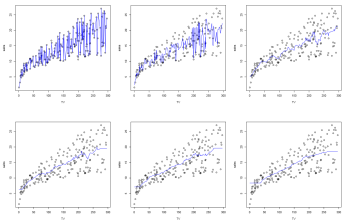
k-Nearest Neighbors: example

We wish to study the performance of KNN for some values of k with the only variable `tv`

k-Nearest Neighbors: example

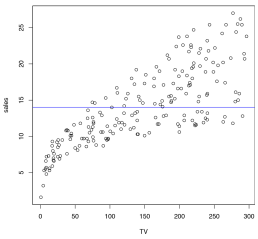
all data

k=1, 2, 10, 20, 30, 50

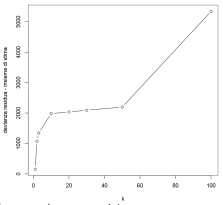


k-Nearest Neighbors: example

k=200



k-Nearest Neighbors: example

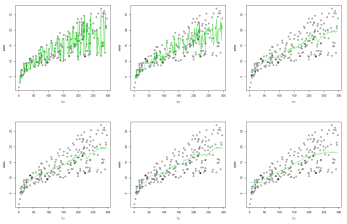


KNN performance decreases as k increases.

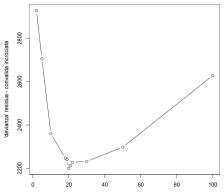
k-Nearest Neighbors: example

Cross validation leave-one-out

k=1, 2, 10, 20, 30, 50



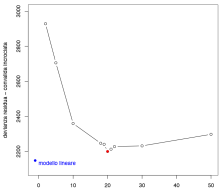
k-Nearest Neighbors: example



A minimum has reached . . .
trade-off between variance and bias

k-Nearest Neighbors: example

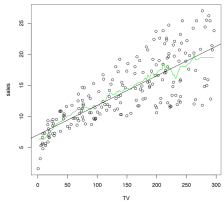
Performance of linear model and KNN with variable `tv`



In the case of `tv` the linear model performs better than the KNN for each value of k.

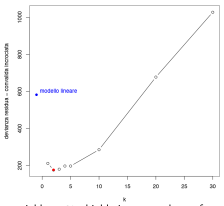
k-Nearest Neighbors: example

Linear model and KNN-20 with variable `tv`



k-Nearest Neighbors: example

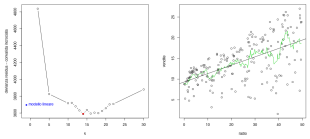
Performance of linear model and KNN with variable `tv` and `radio`



Adding the variable `radio` highly increases the performance of KNN. The minimum is reached for $k = 2$

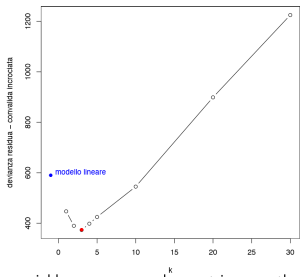
k-Nearest Neighbors: example

Performance of linear model and KNN with variable `radio`



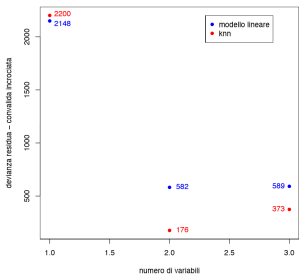
k-Nearest Neighbors: example

Let us add the variable `newspapers`



Adding the variable `newspapers` does not increase the performance of the model.
The KNN is better than the linear model in any case.

k-Nearest Neighbors: example



Chapter 8

Local Regression and Loess

If $f(x)$ is a derivable function in x_0 then, the Taylor's approximation says that it is locally approximated by a line passing through $(x_0, f(x_0))$, i.e.,

$$f(x) = \underbrace{f(x_0)}_{\alpha} + \underbrace{f'(x_0)}_{\beta}(x - x_0) + \text{ error}$$

We introduce the weighted least squares by weighting observations x_i with their distance from x_0 :

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_i - x_0)\}^2 w_h(x_i - x_0)$$

$h(h > 0)$ is a scale factor, called bandwidth or smoothing parameter, and $w_h(\cdot)$ is a symmetric density function around 0, said kernel.
By varying x_0, we obtain a whole estimated curve $\hat{f}(x)$. The most important component is h, which regulates the smoothness of the curve, while the choice of w is less relevant. We could think to w as the density of the normal distribution $N\left(0, h^2\right)$
EXAMPLE.
We can show that the estimate relative to a general point x can be obtained from the explicit formula

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

where $a_r(x;h) = \left\{\sum (x_i - x)^r w(x_i - x;h)\right\}/n$, for $r = 0, 1, 2$ We are therefore dealing with an estimate: explicit, noniterative, linear in y_i; therefore we can write

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

for a suitable vector $s_h \in \mathbb{R}^n$ depending on h.
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

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

where S_h is an $m \times n$ matrix and now x is the vector (in $\mathbb{R}^m$) of x -axis where we estimate function f. 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 of the kernel is not critical, as many studies on the subject have shown. Let $w(t;h) = \frac{1}{h} w_0\left(\frac{t}{h}\right)$. The density $N(0,1)$ is a common choice for w_0, i.e., we choose $N\left(0, h^2\right)$ for $w(t;h)$ Many other choices are possible, in particular those with limited support as e.g. the tricubic or biquadratic ones that is

$$w_0(t) = \begin{cases} \left(1 - t^2\right)^2 & \text{if } |t| < 1 \\ 0 & \text{otherwise} \end{cases} \qquad w_0(t) = \begin{cases} \left(1 - |t|^3\right)^3 & \text{if } |t| < 1 \\ 0 & \text{otherwise} \end{cases}$$

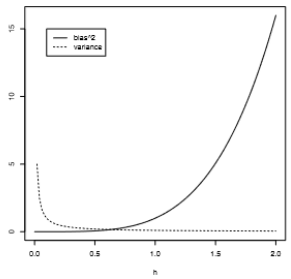
the limited support reduces the computational burden, thanks to the many null terms.
Common choices for kernels:

kernel	$w(z)$	support
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}z^2\right)$	$\mathbb{R}$
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)

A critical aspect is the choice of the **smoothing parameter** h, since we can prove that, for n sufficiently large,

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

where $\sigma_w^2 = \int z^2 w(z)dz, \alpha(w) = \int w(z)^2 dz$ and $g(x)$ indicates the density from which the x_i were drawn; bias is $O\left(h^2\right)$ and variance is $O\left(\frac{1}{nh}\right)$ once again, in choosing h, we have a trade-off between bias and variance.
Bias/variance trade-off



Minimizing bias ^{2} and variance we get

$$h_{\text{Opt}} = \left(\frac{\alpha(w)}{\sigma_w^2 f''(x)^2 g(x) n}\right)$$

however this is useless in practice as $f''(x)$ and $g(x)$ are unknown.
In practice: divide data in training set and test set cross-validation. For leave-one-out, still $y_i - \hat{y}_{-i} = \frac{y_i - \hat{y}_i}{1 - S_{h,ii}}$ information criteria. E.g. the AIC_c results

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

In any case the best that we can obtain is

$$\mathbb{E}\left\{[f(x) - \hat{f}(x)]^2\right\} = O\left(n^{-4/5}\right)$$

which is larger than the one for the parametric model, for which is $O\left(n^{-1}\right)$, if the parametric model is satisfactory. Therefore, nonparametric estimation is intrinsically less efficient than parametric one, if the parametric model is satisfactory.
in many cases, there is an advantage in using a nonconstant bandwidth along the x -axis, according it to the level of sparseness of observed points. **Variable bandwidth**: it is reasonable to use larger values of h when x_i are more scattered $\mathbb{R}^p$ Good idea! ... but how do we modify h? **loess**: express the smoothing parameter defining the fraction of effective observations for estimating $f(x)$ at a certain point x_0 on the x -axis; this fraction is kept constant and this implies automatically a setting of the

bandwidth related to the sparsity of data. In addition, we combine this idea with the use of *robust estimation.*

The usual way to construct confidence intervals for a parameter, starting from an estimator, refers to the pivotal quantity. To develop similar quantity for $f(x)$ we follow the same idea. A pivotal quantity, approximately, is

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

where $b(x)$ indicates the bias. However, for the asymptotically optimal bandwidth, the bias has the same order of magnitude as the denominator. Therefore, the bias term cannot be neglected in this framework, in contrast with what happens in a parametric context.

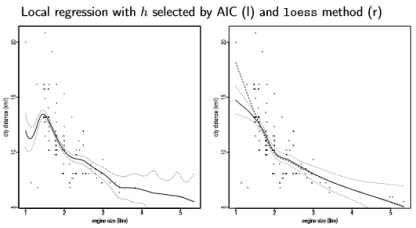
However, instead of looking for extremely complicated corrections, a current solution is to construct variability bands of the type

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

This is clearly an interval, but once the expression is applied to every point on the x -axis, it gives rise to two bands

It provides an indication of the local variability of the estimate. Note that: 1. for every fixed x, these intervals are not confidence intervals, 2. even in the case that $b(x) = 0$, each interval would have been a confidence interval of approximate level $1 - \alpha$ for $f(x)$ only pointwise, for each fixed value of x, but not globally for the entire curve.

Variability bands



8.1 Local Linear Regression in 2D

Consider two predictors, x_1 and x_2, and we presume that

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

where $f(x_1, x_2)$ is now a function from $\mathbb{R}^2$ to $\mathbb{R}$; in our example we take x_1 = engine size, x_2 = curb weight. We want to estimate y in $x_0 = (x_{01}, x_{02})$ the natural extension of the previous criterion takes the form

$$\min_{\alpha,\beta,\gamma} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_{1i} - x_{01}) - \gamma(x_{2i} - x_{02})\}^2 w_{h_1}(x_{1i} - x_{01}) w_{h_2}(x_{2i} - x_{02})$$

where two smoothing parameters, h_1 and h_2, are used, for x_1 and for x_2.

If we indicate by X the $n \times 3$ matrix of which the i -th row is

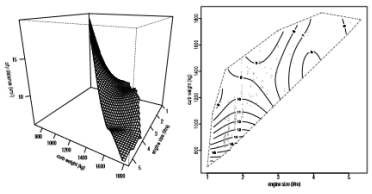
$$\{1, (x_{1i} - x_1), (x_{2i} - x_2)\}$$

$y = (y_1, \ldots, y_n)^\top$ and $W = \text{diag}(w_1, \ldots, w_n)$, then the solution of the previous minimum problem is the first element of

$$\left(X^T W X\right)^{-1} X^T W y$$

this calculation is repeated for every choice of point x_0, and the number of these points should be now higher than in the scalar case

Local linear regression in 2D



Chapter 9

Splines

9.1 Interpolating Splines

'**Spline**' is a mathematical tool useful in many contexts finalised to approximate functions or to *interpolate data*. 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 we look for "smooth" functions between two successive knots, in the interval (ξ_i, ξ_{i+1}), curve $f(x)$ coincides with a suitable polynomial, of prefixed degree d, these sections of polynomials meet at point ξ_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.
In the usual case $d = 3$ and these conditions can be written as

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

where $g\left(x^-\right)$ and $g\left(x^+\right)$ indicate the left and right limits of a function $g(\cdot)$ at point x.
When $d = 3, K$ knots require: 1. $4(K-1)$ parameters 2. K constraints of the type $f(\xi_i) = y_i$ 3. $3(K-2)$ continuity constraints of the function and the first two derivatives. 4. the difference between coefficients and constraints is 2 units, we must therefore introduce two additional constraints. $\rightarrow$ e.g. $f''(\xi_1) = f''(\xi_K) = 0$, natural cubic spline Therefore: **spline = piecewise polynomial functions** (with continuity constraints)

9.2 Regression Splines

We have n observed points (x_i, y_i) for $i = 1, \ldots, n$ that we want to interpolate, but now *not exactly.* We apply these ideas to parametric regression, by fitting a 'cubic spline' to the n points, we divide the x -axis into $K+1$ intervals separated by K knots, $\xi_1, \ldots, \xi_K$, and interpolate the n points with the least squares criterion. The obtained function is called **regression spline**.
We can show that the required piecewise polynomials may be rewritten in the equivalent form

$$f(x; \beta) = \sum_{j=1}^{K+4} \beta_j h_j(x)$$

that is as a suitable linear combination of basis functions.For example in the simple case of two knots, ξ_1 and ξ_2, the basis functions are

$$h_1(x) = 1, \quad h_2(x) = x, \quad h_3(x) = x^2, \quad h_4(x) = x^3$$
$$h_5(x) = (x - \xi_1)_+^3, \quad h_6(x) = (x - \xi_2)_+^3$$

where $(a)_+ = \max(a, 0)$ In general, if we have K knots, the basis functions will be $K + 4$ functions $h_j(x)$
Thus we can write

$$f(x_i; \beta) = \sum_{j=1}^{K+4} \beta_j h_j(x_i) = H(x_i)\beta$$

where $\beta = (\beta_1, \ldots, \beta_{K+4})^\top$ & $H(x_i) = \{h_1(x_i), \ldots h_{K+1}(x_i)\}^\top$ are both $K + 4$ -size vectors. β represents a vector of unknown basis coefficients and $H(x_i)$ the basis expansion of x_i. Now let $y = (y_1, \ldots, y_n)^\top$ and $\mathbf{H} = \left(H(x_1)^\top, \ldots, H(x_n)^\top\right)$ Then use the OLS and get

$$\hat{\beta} = \arg\min_{\beta} \left\{ \|y - \mathbf{H}\beta\|_2^2 \right\}$$

since H is nothing but a matrix we have

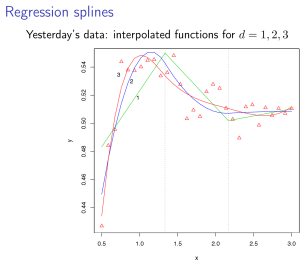
$$\hat{\beta} = \left(\mathbf{H}^\top \mathbf{H}\right)^{-1} \mathbf{H}^\top y$$

as in the standard linear regression. We call this estimator a linear estimator,Note that we have

$$\mathbf{H}\left(\mathbf{H}^\top \mathbf{H}\right)^{-1} \mathbf{H}^\top = S_K$$

which we call smoothing matrix keeping in mind that it is nothing but a projection matrix Indeed

$$\hat{y} = S_K y$$



9.3 Smoothing Splines

Let us consider the **penalized least squares criterion**

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

where λ is a positive penalisation parameter of the roughness degree of curve f (quantified by the integral of $f''(x)^2\Big)$, and therefore acts as a smoothing parameter.

A noteworthy mathematical result shows that the solution to that minimization problem is represented by a natural cubic spline whose knots are distinct points x_i :

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

where n_0 is the number of distinct x_i and the $N_j(x)$ are natural cubic splines basis functions.
Note that differently from the regression splines, here the number of parameters and of knots are the same. This is true given the constraints induced by the natural cubic spline We can rewrite the penalized least squares criterion

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

as

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

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 optimisation problem is given by

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

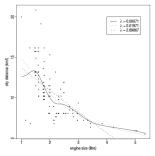
which depends on the choice of the smoothing parameter λ. Thus we can write

$$\hat{y} = S_\lambda y$$

for a certain matrix S_λ of dimension $n_0 \times n_0$. We are dealing with another linear estimator. These are called **smoothing splines**.

Smoothing splines

Car data: Estimate of `city distance` according to `engine size` by a smoothing spline, for three choices of λ



We can also use the criteria discussed earlier for the choice of smoothing parameter λ

Chapter 10

Generalised Addictive Models

10.1 Effective Degrees of Freedom

From the theory of the linear models we know that

$$\hat{y} = Py, \quad \hat{\varepsilon} = (I - P)y$$

where $P = X\left(X^\top X\right)^{-1} X^\top$ is a symmetric idempotent matrix of rank p. We also know that the rank is equal to $\mathrm{tr}(P)$, the trace of P and $E\left(\|\hat{\varepsilon}\|^2\right) = \sigma^2(n-p)$ From here, adding the Gaussian hypothesis, we have a number of properties related to the associated quadratic forms, for example

$$\|\hat{y}\|^2 = \hat{y}^\top \hat{y} \sim \sigma^2 \chi^2_p(\delta), \quad \|\hat{\varepsilon}\|^2 \sim \sigma^2 \chi^2_{n-p}$$

where δ is a non centrality parameter
Total variance $\|y\|^2 = y^\top y$ can be decomposed in two components (error deviance, $\|\hat{\varepsilon}\|^2$, and regression deviance $\|\hat{y}\|^2$:

$$\|y\|^2 = \|\hat{\varepsilon}\|^2 + \|\hat{y}\|^2$$
$$\text{Total SQ} = \text{ Residual SQ} + \text{ Model SQ}$$

we can therefore fill the **analysis of variance** table. The sum of squares of regression, $\|\hat{y}\|^2$, can also be decomposed in individual components, one for each explanatory variable, with a corresponding decomposition of the degrees of freedom.
Most of the nonparametric methods described so far are linear forms of the response variable

$$\hat{y} = S_h y, \quad (S_h \,\text{matrix}\, n \times n)$$

the corresponding vector of the residuals is $\hat{\varepsilon} = (I - S_h)\, y$. Even if we assume the normality of ε, the probability distribution of $\|\hat{\varepsilon}\|^2$ is no longer χ^2. However, we have empirical evidence based on simulations indicating that the shape of the probability density for $\|\hat{\varepsilon}\|^2$ is similar to that of χ^2.
$\|\hat{\varepsilon}\|^2 \sim$**approx** $\sigma^2 \chi^2_?$ $\|\hat{y}\|^2 \sim$ approx $\sigma^2 \chi^2_{??}(\delta)$
We look for an expression that plays the role of degrees of freedom. Consider the residual sum of squares

$$Q = \sum_i \hat{\varepsilon}_i^2 = \|\hat{\varepsilon}\|^2 = \hat{\varepsilon}^\top \hat{\varepsilon} = y^\top (I_n - S)^\top (I_n - S)\, y$$

in view of the correspondence between the average value and degrees of freedom for a χ^2, we obtain the expected value

$$\mathbb{E}\{Q\} = \mathbb{E}\left\{Y^\top (I_n - S)^\top (I_n - S)\, Y\right\} =$$
$$= \mu^\top (I_n - S)^\top (I_n - S)\, \mu + \sigma^2 \,\mathrm{tr}\, (I_n - S)^\top (I_n - S)$$

where $\mu = \mathbb{E}\{Y\}$, and we use the following Lemma.

Lemma 10.1.1. *Let $A = (a_{ij})$ a square matrix of order p. Therefore*

$$\mathbb{E}\left\{X^\top AX\right\} = \mu^\top A\mu + \mathrm{tr}(AV)$$

where $\mu = \mathbb{E}\{X\}$ and $V = \mathrm{var}\{X\}$

From

$$\mathbb{E}\{Q\} = \mu^\top (I_n - S)^\top (I_n - S)\,\mu + \sigma^2 \,\mathrm{tr}\,(I_n - S)^\top (I_n - S)$$

by introducing the approximations

$$(I_n - S)\,\mu \approx 0, \quad (I_n - S)^\top (I_n - S) \approx (I_n - S)$$

we can then write

$$\mathbb{E}\{Q\} \approx \sigma^2\{n - \mathrm{tr}(S)\}$$

We call - $\{n - \mathrm{tr}(S)\}$ effective or equivalent degrees of freedom for the error term - $\mathrm{tr}\,(S)$ effective degrees of freedom for the smoother. We can introduce slightly different expressions for the same degrees of freedom, based on alternative approximations, e.g.

$$\mathrm{tr}\left(SS^\top\right) \quad \text{or} \quad \mathrm{tr}\left(2S - SS^\top\right)$$

We have relaxed the linearity assumption while still attempting to maintain as much interpretability as possible. To this end, we consider approaches such as splines, local regression, and generalized additive models.

Regression splines involve dividing the range of X into K distinct regions. Within each region, a polynomial function is fit to the data. However, these polynomials are constrained so that they join smoothly at the region boundaries, or knots. Provided that the interval is divided into enough regions, this can produce an extremely flexible fit. Smoothing splines are similar to regression splines, but arise in a slightly different situation. Smoothing splines result from minimizing a residual sum of squares criterion subject to a smoothness penalty. Local regression is similar to splines, but differs in an important way. The regions are allowed to overlap, and indeed they do so in a very smooth way. Generalized additive models allow us to extend the methods above to deal with multiple predictors.

10.2 Generalised Additive Models

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

Consider

$$f\,(x_1, \ldots, x_p) = \alpha + \sum_{j=1}^{p} f_j\,(x_j) + \sum_{k<j} f_{kj}\,(x_k, x_j) + \ldots$$

truncate the series to first (or second) order terms.This implies a limitation to the form of $f\,(x_1, \ldots, x_p)$, but still allows for a lot of flexibility. In the case of stopping at the first order, we are considering

$$y = f\,(x_1, \ldots, x_p) + \varepsilon = \alpha + \sum_{j=1}^{p} f_j\,(x_j) + \varepsilon$$

where the p functions $f_j\,(x_j)$ are estimated by the **backfitting** algorithm.

Note that, in order to avoid an overparametrisation of the intercept, each f_j needs to be of 'zero mean'.

We may write

$$f_j\,(x_j) = y - \left(\alpha + \sum_{k \neq j} f_k\,(x_k) + \varepsilon\right)$$

Backfitting algorithm

Initialisation: $\hat{\alpha} = \sum_i y_i/n, \hat{f}_j = 0, \forall j$ Cycle: for $j = 1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$

$$\hat{f}_j \leftarrow \mathcal{S}\left[\left\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k\left(x_{ik}\right)\right\}_1^n\right]$$

$$\hat{f}_j \leftarrow \hat{f}_j - n^{-1} \sum_{i=1}^{n} \hat{f}_j\left(x_{ij}\right)$$

until the functions $\hat{f}_j$ are stable.

$$y = f_1\left(x_1\right) + f_2\left(x_2\right) + \cdots + f_p\left(x_p\right) + \varepsilon$$

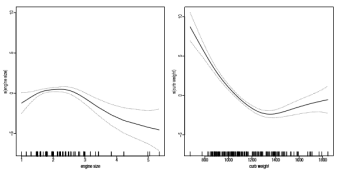
Backfitting Algorithm

$$\begin{aligned} f_1\left(x_1\right) =& S_1\left(y - \cdot - f_2\left(x_2\right) - \cdots - f_p\left(x_p\right)\right) \\ f_2\left(x_2\right) =& S_2\left(y - f_1\left(x_1\right) - \cdot - \cdots - f_p\left(x_p\right)\right) \\ & \vdots \\ f_p\left(x_p\right) =& S_p\left(y - f_1\left(x_1\right) - f_2\left(x_2\right) - \cdots - \cdot\right) \end{aligned}$$

where S_j are: univariate regression smoothers such as smoothing splines, loess, kernel linear regression operators, yielding polynomial fits, piecewise constant fits, parametric spline fits, etc. more complicated operators such as surface smoothers for second order interactions.

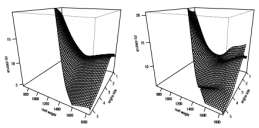
Additive models: Example

Estimate of city distance according to engine size and curb weight by an additive model with a spline smoother



Example

The price for additivity



Left plot: for each `engine size`, the function of `curb weight` has the same shape — the level differs, and vice versa.
Right plot: a two-dimensional smoother appears to have found an interaction (turns out not to be significant).

Additive models: Example

Analysis of variance

component	deviance	df	p-value
engine size	1169618	12.7	0.000
curb weight	729	5.40	0.094
engine size, curb weight	410.2	13.08	

We may approximate the distribution of a test F with a Snedecor F with $(tr(S) - tr(S_0), n - tr(S))$ degrees of freedom.

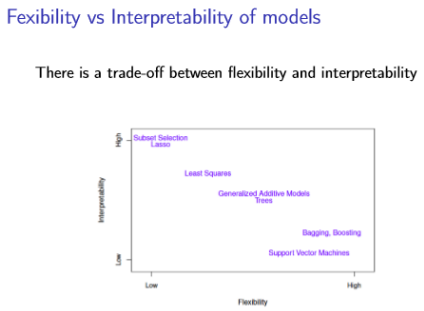
$$F = \frac{729/5.40}{410.2/(203 - 13.08)}$$

GAM important properties:

- 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.
- The non-linear fits can potentially make more accurate predictions for the response Y.
- 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.
- If we are interested in inference, GAMs provide a useful representation.

Chapter 11

Gradient Boosting



11.1 Boosting

Initially developed for classification problems, later extended to regression problems. Idea: assign more weight to observations badly classified, to make the model work more on these → **AdaBoost** Bagging, Boosting and Random Forests use trees as building blocks to construct more powerful models.

11.2 Gradient Boosting

Powerful algorithm of machine learning. Employed for both regression and classification problems. **Gradient Boosting = Gradient Descent + Boosting**.
Let us consider a simple regression problem ... with a simple case: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ 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: $f(x_1) = 0.8$ while $y_1 = 0.9, f(x_2) = 1.4$ while $y_2 = 1.3$ How can we improve the model?
Consider that: - 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

$$y_i = f(x_i) + h(x_i)$$

The prediction is updated as follows:

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

But we can also write

$$y_1 - f(x_1) = h(x_1)$$
$$y_2 - f(x_2) = h(x_2)$$
$$\vdots$$
$$y_n - f(x_n) = h(x_n)$$

where $r(x_i) = y_i - f(x_i)$ are the **residuals**.
Gradient Boosting $\rightarrow$ fit a regression tree, h, on data $(x_1, r_1), (x_2, r_2), \ldots, (x_n, r_n)$ 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 repete 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 repeate 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*?
~~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 $J = \frac{1}{n}\sum_i L(y_i, f(x_i))$

$$\frac{\partial J}{\partial f(x_i)} = \frac{\partial \sum_i \frac{1}{n} L(y_i, f(x_i))}{\partial f(x_i)} = \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} = f(x_i) - y_i$$

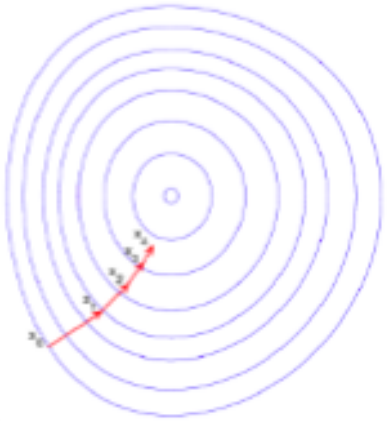
We can see the *residuals as negative gradients*

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

11.2.1 Gradient Descent

Minimizes a function going in the opposite direction with respect to the gradient

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



How is this related to the Gradient Descent? For a regression problem with quadratic loss function, - residual $\leftrightarrow$ negative gradient - fit h to the residual $\leftrightarrow$ fit h to the negative gradient - update f through the residual $\leftrightarrow$ update f through the negative gradient *We are using the negative gradient*.

11.2.2 Gradient Boosting: Algorithm

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

- initialize the model with a constant value

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

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

- at each iteration $1 < m < M$ calculate the negative gradients for $i = 1, 2, \dots, n$

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

- estimate a regression tree $h_m(x)$ on $-g(x_i)$ giving terminal regions $R_{jm}, j = 1, 2, \dots, J_m$
- for $j = 1, 2, \dots, J_m$ calculate

$$\gamma_{jm} = \arg\min \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

- update the model $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
- Output: $\hat{f}(x) = f_M(x)$

Note: we use the negative gradients because we can use loss functions other than the quadratic loss and derive the corresponding algorithms.
Why should we use different loss functions? Quadratic loss function is: simple to handle mathematically ... not robust with respect to outliers

y_i	0.5	1.2	2	5^*
$f(x_i)$	0.6	1.4	1.5	1.7
$L(y-f)^2/2$	0.005	0.02	0.125	5.445

$\rightarrow$ The presence of an outlier may have negative effects on the general performance of the model.
Other loss functions absolute loss function $L(y, f) = |y - f|$ Huber loss function $\rightarrow$ more robust with respect to outliers $L(y, f) = \begin{cases} 1/2(y-f)^2 & |y-f| \leqslant \delta \\ \delta(|y-f| - \delta/2) & |y-f| > \delta \end{cases}$

y_i	0.5	1.2	2	5^*
$f(x_i)$	0.6	1.4	1.5	1.7
quadratic	0.005	0.02	0.125	5.445
absolute	0.1	0.2	0.5	3.3
Huber ($\delta = 0.5$)	0.005	0.02	0.125	1.525

11.2.3 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. **Shrinkage:** The update rule is modified in this way

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

Parameter $0 < \nu < 1$ controls the 'learning rate' of the boosting procedure. Smaller values of $\nu \rightarrow$ more shrinkage $\rightarrow M$ bigger Trade-off between ν and M.

Why Gradient Boosting? - use of 'mixed' data - robust to outliers in input - interpretability of results - prediction power

Gradient Boosting

Comparison among models

MART → Gradient Boosting

Some characteristics of different learning methods.
Key: ●= good, ●=fair, and ●=poor.

Characteristic	Neural Nets	SVM	CART	GAM	KNN, kernels	MART
Natural handling of data of "mixed" type	●	●	●	●	●	●
Handling of missing values	●	●	●	●	●	●
Robustness to outliers in input space	●	●	●	●	●	●
Insensitive to monotone transformations of inputs	●	●	●	●	●	●
Computational scalability (large N)	●	●	●	●	●	●
Ability to deal with irrelevant inputs	●	●	●	●	●	●
Ability to extract linear combinations of features	●	●	●	●	●	●
Interpretability	●	●	●	●	●	●
Predictive power	●	●	●	●	●	●

Gradient Boosting: example

- Data set on house prices in California
- y= median price in hundreds of thousands dollars
- demographic variables: average income (MedInc), house density (House), average number of people per house (AveOcc), population (Population)
- house features: latitude, longitude (latitude, longitude), average number of rooms (AveRooms) average number of bedrooms (AveBedrms), age of the house (HouseAge)
- 8 variables

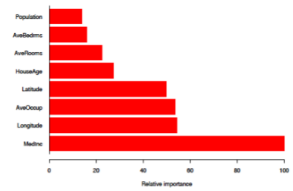
Gradient Boosting: example

Visualizing results

- relative influence plot: reduction in squared error due to each variable

Gradient Boosting: example

Gradient Boosting with tree depth= 6, shrinkage= 0.1, loss function= Huber



Case study

We are in New York in 2019
we want ot understand what are the factors determining prices on Airbnb ...

Open Data in New York

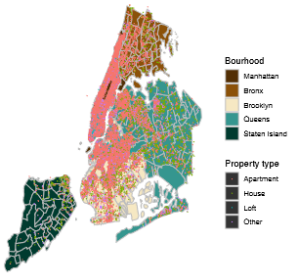
From the website `https://opendata.cityofnewyork.us/` we may collect information referring to:

variables	*source*
major attractions	Dept Finance
hotels	NYC open data
restaurants	NYC open data
metro	Metro trans authority
spare time	NYC open data
helath services	NYC open data
crime	NY Police Dept

...and much more.

Airbnb in New York

Property position



Case study

- What are the drivers of house prices?
- Hedonic price modelling: house prices depend on their characteristics (size, number of rooms, type of house ...)
- anything else? we may think that there are other factors giving value to a house, and therefore to its price.

Case study

We would like to account for 3 major points:

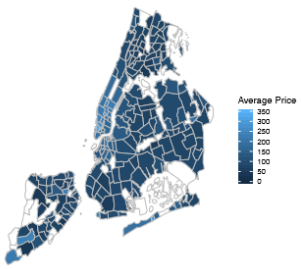
- Airbnb diffusion
- heterogeneous districts
- availability of Open Data

Airbnb in New York

- The website `insideairbnb.com` contains the 'listings' of Airbnb houses for many cities around the world.
  For each listing, many variables are available referring to
  - property,
  - host,
  - *guest reviews*,
  - terms of service.
- of course there is also the variable 'price per night'
- what are the variables that play a major role in price determination? ...

Airbnb in New York

Average price of houses



Case study

The 'hedonic' approach has been used to explain differences in hotel prices ... and more recently to study the phenomenon of Airbnb in different cities around the world

- Airbnb: example of sharing economy
- Business opportunity or threat?
- Unfair competition towards hotels?
- Increasing rent prices?
- What are the factors determining Airbnb house prices?

Open Data: why?

We would like to account for 'external' information
Open Data are

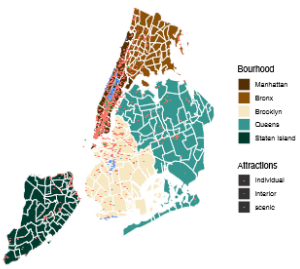
- accessible
- available
- integrable
- updated periodically
- machine readable

Airbnb in New York

- We also want to take care of Open Data
- crime rate, distance from touristic attractions, metro stations ...do they have a role?
- Data Integration

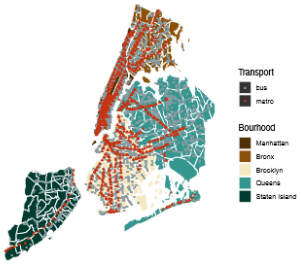
Airbnb in New York

Main attractions



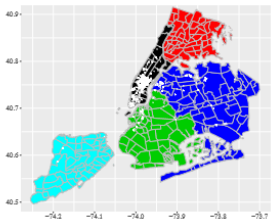
Open Data in New York

Public transport



Open Data in New York

Hotel position



Gradient Boosting for Airbnb prices

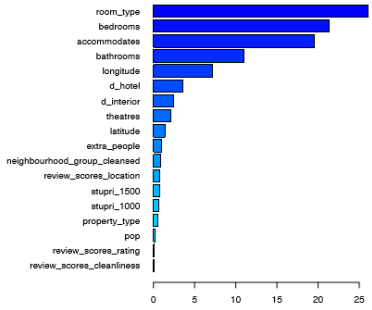
Gradient Boosting for Airbnb prices

Airbnb house prices in New York with GB

- large dataset: 77000 obs, 107 variables
- response: price/night
- training set: 50000
- initial model
  iterations= 100, tree depth= 1 (stump), shrinkage= 0.1
- other options are possible by modifying tuning parameters

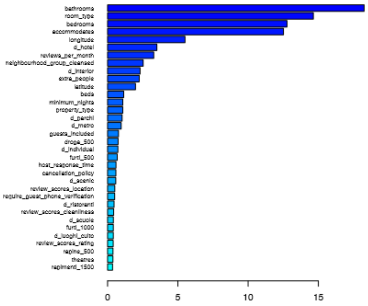
Gradient Boosting for Airbnb prices

iterations= 100, depth= 1, shrinkage= 0.1



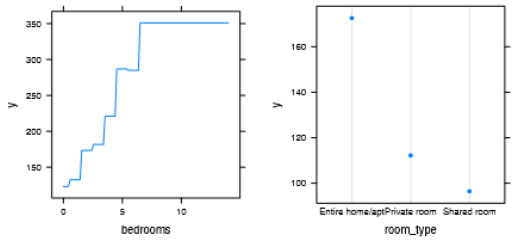
Gradient Boosting for Airbnb prices

iterations= 180, depth= 4, shrinkage= 0.2

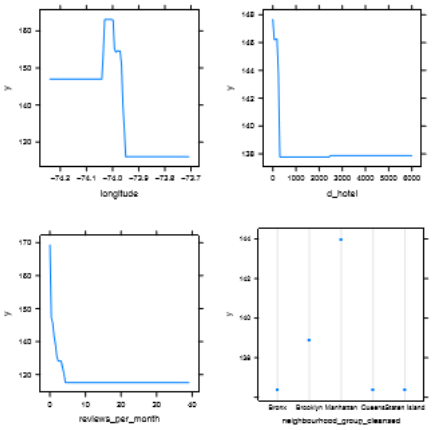


Gradient Boosting for Airbnb prices

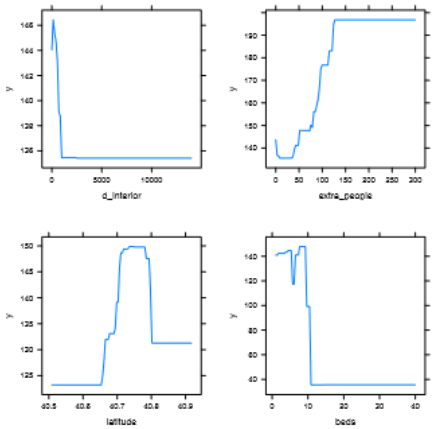
Partial dependence plots: illustrate the marginal effect of the selected variables on the response after integrating the other variables.



Gradient Boosting for Airbnb prices



Gradient Boosting for Airbnb prices



Gradient Boosting for Airbnb prices