STAT 443: Forecasting

Paul Marriott

December 29, 2015

Contents

1

Introduction to forecasting, control and time series

> "There are two kinds of forecasters: those who don't know, and those who don't know they don't know." J. K. Galbraith

1.1 Introduction

This set of notes draws from a number of resources and details can be found in the bibliography. The material in Chapter 2 – on regression and model building for forecasting – is strongly influenced by the book Hastie et al. (2009, Ch. 1, 3 and 7). Another good general reference to regression modelling is Abraham and Ledolter (2006). The stationary process material in Chapter 3 and its application to Box-Jenkins is treated in Box and Jenkins (1976), Brockwell and Davis (2002) and Brockwell and Davis (2009). The second of these is perhaps most suitable for the novice. Another very good time series book is Shumway and Stoffer (2010). The Bayesian material in Chapter 5 is motivated by West and Harrison (1997) with corresponding state space modelling in Brockwell and Davis (2002). Other interesting books include Tong (1990) on non-linear time series and Lamigueiro (2014) on displaying time series.

1.2 Examples

1.2.1 Observed data examples

We start with examples of *time series* which form an important basic form of data from which we perform forecasts. Issues illustrated include the types of structure that we commonly come across with this kind of data. In particular we note the following key points: (i) unlike examples from earlier statistics courses the data does not come as independent realisations

of a random variable, (ii) the data is not identically distributed where both the mean and, potentially, the variance can vary in time, (iii) there can be abrupt changes in the data generation process.

Example 1.2.1. (Actuarial Science) Insurance companies offering accident insurance need to have forecasts of the number of payouts they are likely to have in a given period in order to set premiums. Of course they also need to have forecasts on severity of the accident but for simplicity lets focus on just the number of cases.

Illustrated in Fig. 1.1 are monthly number of deaths and serious injuries in UK road accidents and Ontario respectively, (R Core Team, 2013). The time ranges are from January 1969 to December 1984 and January 1960 to December 1974. For the UK data, the description of the dataset mentions that a seatbelt law was introduced in February 1983.

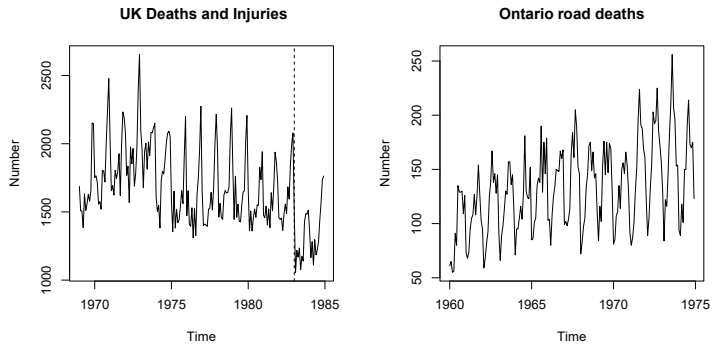

Figure 1.1: Number of deaths and serious injuries in road accidents per month

Fig. 1.1 are examples of *time series* plots.

Definition 1.2.2. (Time series plot) A time series plot is a scatterplot of the data with time on the x-axis in, typically, equally spaced intervals, and the observations on the y-axis. For visual clarity adjacent observations are connected by lines. The equally spaced intervals can be in units of years, months, days, hours etc and these units are called the *period* of the time series.

In Example 1.2.1 within each year there is a repeated pattern across the months of the year.

Definition 1.2.3. (Level, trend and seasonal component) Terms like *level*, *trend* and *seasonality* can only be defined precisely in terms of a modelling exercise. Here we give informal definition for use in describing observed

patterns in time series plots. The *level* is the local mean of the observations, around which we see random *noise*. When the level varies with time we say there is a *trend*. A *seasonal effect* is a systematic and calendar related effect which repeats with a given period.

In the Ontario accident data we notice that, on the yearly time scale after smoothing the seasonal aspects, that there is a steady increase in the average number of accidence across the time period, the trend. A final feature of this data is the possibility of a *change point*, perhaps due to the change in law in the UK data

Definition 1.2.4. (Change point) A change point is a time at which at least of the of the following changes: the data generation process, the way that the data is measured, or the way that the observation is defined.

Definition 1.2.5. (Non-stationarity) Seasonality, trends, non-constant variance and change points are all examples of *non-stationarity*. Stationarity, defined formally in Chapter 3, informally means that the underlying random process does not change in time.

If we have a stationary process we would be able to assume that, at least statistically, the future is similar to the past – this means that we could perhaps use information from the past to make forecasts about the future. Exact stationarity is rare and we need to be to model the forms of non-stationarity seen in the data.

Example 1.2.6. (Ontario Gas prices) Figure 1.2 shows the monthly demand for gasoline in Ontario 1960 to 1975 in dollars. We see strong seasonality in the data as well as a non-linear change in the level. Underlying forecasting problems here might have been, for an analyst in 1975, to forecast gas usage in the upcoming year.

Example 1.2.7. (Airline passengers) In Fig. 1.3, which shows the monthly totals of international airline passengers from 1949 to 1960, (Box and Jenkins, 1976), we see a strong trend and a seasonal pattern. This pattern is a little different from that of Example 1.2.6 since the size of the periodic oscillations increases with time. We might model this as having the periodic seasonal effect being multiplicative with the time series, i.e. a proportional change.

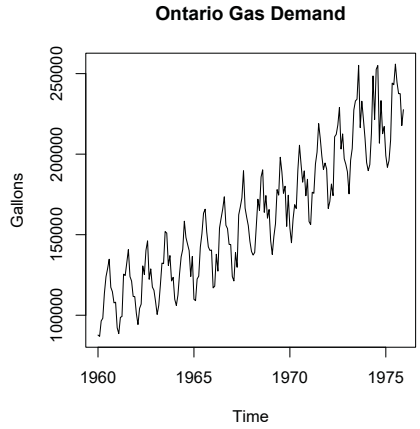

Figure 1.2: Monthly demand for gasoline in Ontario from 1960 to 1975

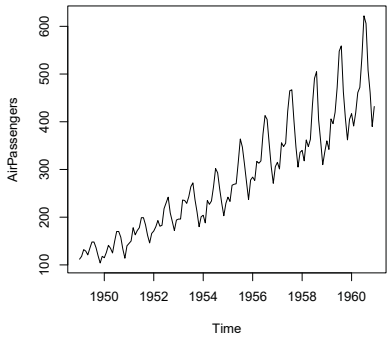

Figure 1.3: Monthly totals of international airline passengers, 1949 to 1960.

Example 1.2.8. (Internet network traffic) Figure 1.4 are plots of time series of the network traffic in the UK academic network backbone[1]. Data was collected between 19 November 2004 and 27 January 2005, at the frequency daily and hourly time scales. In each time period the number of *bits* has been recorded. We see different aspects of the data on these different time scales.

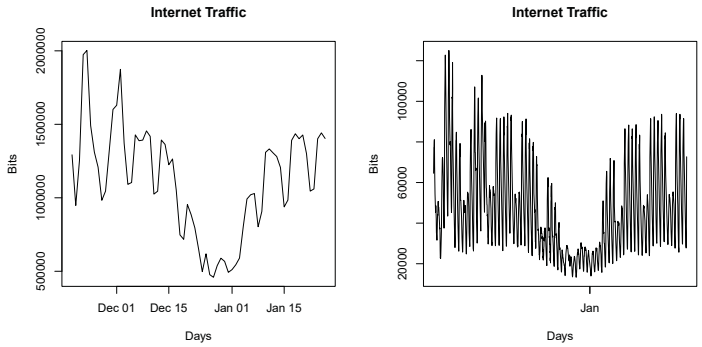

Figure 1.4: Network traffic in the UK academic network daily and hourly intervals.

Example 1.2.9. (Stock price) Figure 1.5 shows an example of daily stock prices, from the FTSE index in the period 1991-1996, (R Core Team, 2013). The right hand plot shows the first difference of the stock price, i.e. $log(X_t/X_{t-1})$ if X_t is the time series. This is a measure of the relative daily change in the price. We see that the variability of this seems to change with time.

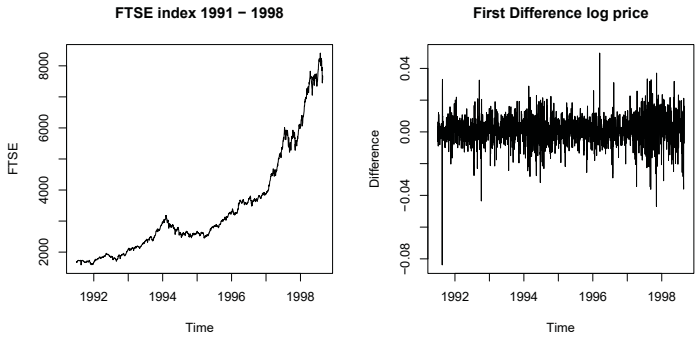

Figure 1.5: Daily FTSE stock price.

[1]https://datamarket.com/

Example 1.2.10. (US unemployment) An example from economics is shown in Fig. 1.6, the monthly numbers of registered unemployed in the US, (R Core Team, 2013). We see periods where the level is low and periods where it is high. This is representative of the so-called *business cycle*, (Sherman and Kolk, 1996). This a cycle of expansions and contractions in the economy that occur at about the same time in many economic activities. They are recurrent but not strictly periodic with durations varying from one to twelve years.

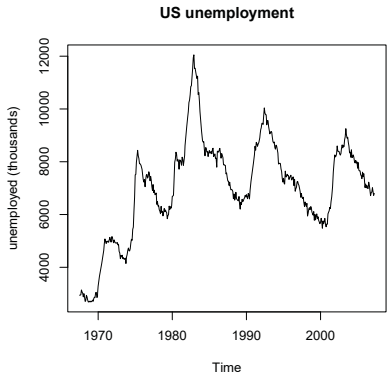

Figure 1.6: US unemployment

Example 1.2.11. (Denmark birth data) The data illustrated in Fig. 1.18 shows the number of births in Denmark for each month in the period 1900-1992. The data shows strong seasonality, non-constant level and at least one change point at 1919.

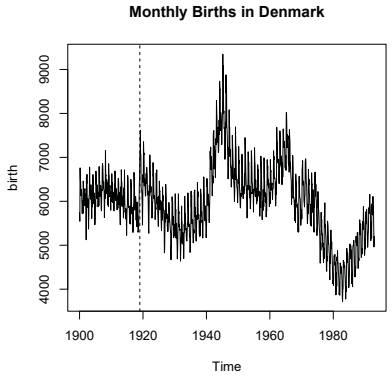

Figure 1.7: Monthly birth data in Denmark 1900-1992

1.2.2 Mathematical models

We can illustrate some important issues related to time series data from purely mathematical models.

Example 1.2.12. (Iterative maps) A simple iterative dynamical system defined by

$$x_{n+1} = F_\lambda(x_n),$$

where $F_\lambda(x) := \lambda x(1-x)$. We can think of time as being $n \in \{1, \ldots, N\}$ and each observation depends, deterministically on the previous value. Consider the forecasting problem of forecasting x_{n+h} where we had observed $x_1, \ldots, x_n$ and the function F_λ. It would seem that the forecasting problem

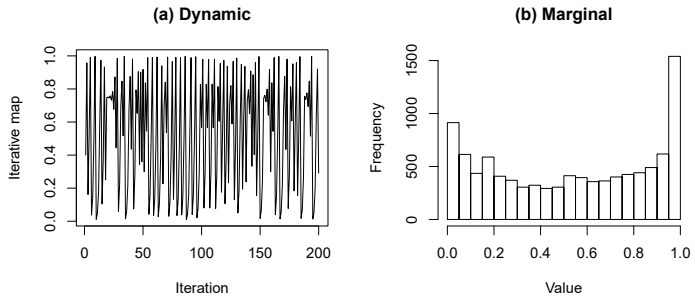

Figure 1.8: Iterative maps.

here is straightforward since this deterministic. However this is an example of a *chaotic system* in which any small change in the value of x_n is reflected in very large changes in x_{n+h} – the so-called butterfly effect. This time series is thus pseudo-random or chaotic. Such time series are studied in Tong (1990).

What can we learn from such an example? First, that it is not clear that we can detect stationarity just by looking at a time series plot. In Panel (a) we see places where the variance seems to be small, places where it is large. We might then say this looks non-stationary since the variance seems to depend on time, but of course this is a deterministic function which does not change in time so is stationary, in some sense.

Anther important general question is, for large $h > 0$ how can we predict the value of the time series at time $n+h$ if we have observed it up to time n? Since the system is chaotic the past if forgotten very quickly. We can still make forecasts though, for example we could give the distribution shown in Panel (b). This is the so-called equilibrium, or long run, distribution. In this case it is a β-distribution and the past data could be used to estimate its parameters.

Definition 1.2.13. (Markov chain) An example from probability theory of a discrete state time series is a Markov chain, a form of stochastic iteration. It is a sequence X_i of dependent random variables where the distribution of X_{n+1} depends on X_n and only on X_n. That means

$$P(X_{n+1}|X_0 = x_0, \cdots, X_n = x_n) = P(X_{n+1}|X_n = x_n), \tag{1.1}$$

the *Markov property*. As with any iterative scheme you need to define the initial conditions. The initial value X_0 can be chosen deterministically or it can be random with some chosen distribution.

If X_n takes values in the finite set $S = \{1, \ldots, K\}$ it is a *discrete space Markov chain* and

$$P\{X_n = j|X_{n-1} = i\} := p_{ij},$$

defines the $K \times K$ *transition matrix*. The n-step probabilities $p_n(i, j)$ are defined as follows:

$$p_n(i, j) = P\{X_n = j|X_0 = i\} = P\{X_{n+k} = j|X_k = i\},$$

and it can easily be shown that n-step transition probability is the i, jth entry of the matrix P^n. For large n the n-step transition probability often converge to equilibrium distribution which is independent of the past.

Example 1.2.14. (Simple Markov chain)

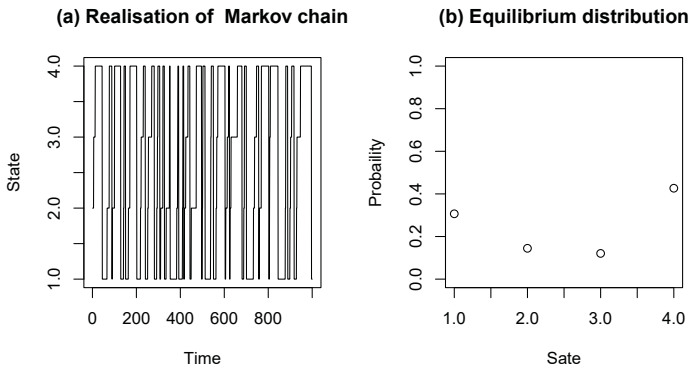

Figure 1.9: Realisation of Markov chain

Figure 1.9 (a) shows a realisation of a 4-state Markov chain which has the transition matrix

$$P = \begin{pmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.9 & 0.1 \\ 0.1 & 0.0 & 0.0 & 0.9 \end{pmatrix}$$

and (b) shows its equilibrium distribution.

We might think about forecasting in Example 1.2.14 based on data. To make a short term forecast you need the current state, X_n, and an estimate of the transition matrix which might be based on all available data. Of course we are here making a modelling assumption that what we have seen is a realisation of a stationary Markov chain. A second, conceptual point, is to note that Fig. 1.9 (a) only shows a single realisation of the Markov chain although we see 1000 observations. We have to be careful with the concept of *sample size* in any situation where we have dependent data.

Example 1.2.15. (Random walk) Consider a continuous state space Markov chain. Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of random variables. The series defined by

$$X_t := \sum_{i=1}^{t} Z_i,$$

for $t = 1, 2, \ldots$, is called a *random walk*.

A related Markov process has the following definition.

Example 1.2.16. (Autoregressive model) Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of $N(0, \sigma^2)$ random variables. The series defined by

$$X_t = \phi X_{t-1} + Z_t$$

for all $t \in \mathbb{Z}$, is called *a first order autoregressive process* (AR(1)) *process*.

Not all time series have to be on discrete time points. If the time period of observation is arbitrary then we talk about *continuous time stochastic processes*

Definition 1.2.17. (Stochastic process) A *stochastic process* is a family of random variables $\{X_t, t \in T\}$ defined on a probability space $(\Omega, \mathcal{F}, P)$ i.e a sample space, a set of events and a probability function. The *index set* T could be a function of the reals or something more general. The functions $\{X_{(\cdot)}(\omega), \omega \in \Omega\}$ on T are known as the realisations or sample paths of the process $\{X_t, t \in T\}$

When the index set T is not finite we can't just write down a joint distribution to define the behaviour of the random variables. The following definition shows what we can do when we look at finite subsets of T.

Theorem 1.2.18. (Kolmogorov's Existence Theorem) The probability distribution functions $F_t(\cdot)$ are the distributions functions of some stochastic process if and only if for any $n \in \{1, 2, \ldots\}$ and $t = (t_1, \ldots, t_n) \in \mathcal{T}$ and $1 \leq i \leq n$,

$$\lim_{x_i \to \infty} F_t(x) = F_{t(i)}(x(i))$$

where $t_{(i)}$ and $x_{(i)}$ are the $(n-1)$-component vectors obtained by deleting the i^{th} components of t and x respectively.

Example 1.2.19. (Gaussian process) Let $T = \mathbb{R}$, then $\{X_t\}$ is a discrete Gaussian process if any finite subset $\{X_{t_1}, \ldots, X_{t_n}\}$ has an n-dimensional multivariate normal distribution. This model is completely determined when the mean and variance-covariance structures are known.

Figure 1.10 shows three realisations from three different one dimensional Gaussian processes. They all share the fact that the realisation gives a continuous graph but they differ in the 'smoothness' of the realisation. This smoothness is controlled by a single parameter ν and the amount of 'smoothness' increases as we go from left to right.

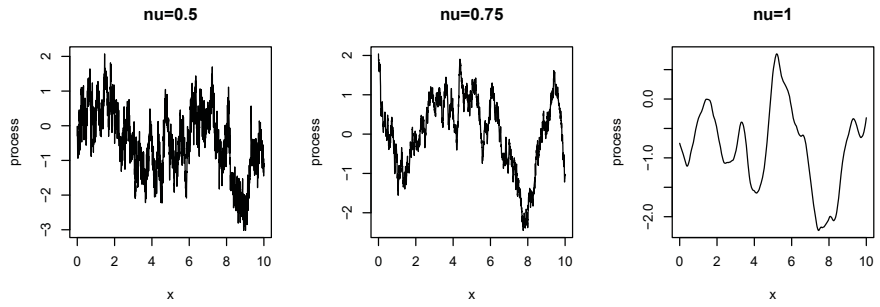

Figure 1.10: Three realisation of one dimension Gaussian processes

Example 1.2.20. (Brownian motion) A *Brownian motion* or *Wiener process* is a continuous-time stochastic process which is the unique process W_t which satisfies the following:

1. $W_0 = 0$,
2. The function $t \to W_t$ is almost surely everywhere continuous,
3. The increments $W_{t_1} - W_{s_1}$ $W_{t_2} - W_{s_2}$ are independent when $0 \leq s_1 < t_1 \leq s_2 < t_2$,
4. The increment $W_t - W_s$ has a $N(0, t - s)$ distribution for $0 \leq s < t$.

We can think of the Brownian motion as a limit of the random walk defined in Example 1.2.15 when time intervals shrink to zero, and we illustrate a realised path in Fig. 1.11(a).

Example 1.2.21. (O-U process) The *Ornstein-Uhlenbeck* process can be thought of as a scaled Brownian motion defined by

$$X_t = \mu + \frac{\sigma}{\sqrt{2\theta}} \exp(\theta t) W_{\exp(2\theta t)}.$$

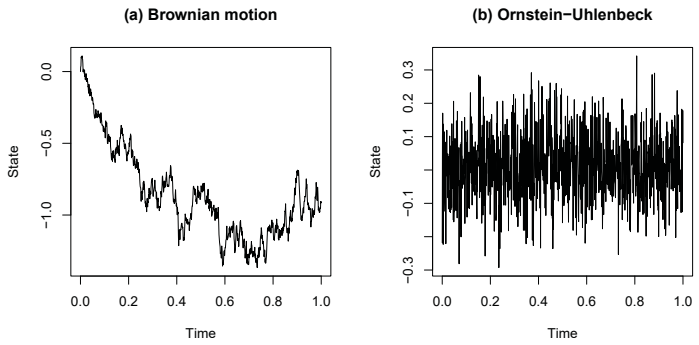

Figure 1.11: (a) Realisation of Brownian motion (b) Realisation of Ornstein-Uhlenbeck process

We illustrate a realised path in Fig. 1.11(b). Such a process can be used for modelling interest rates Vasicek (1977).

Example 1.2.22. (Non-constant variance process) If we want models where the underlying variance of the process changes with time then one possibility is to used a AutoRegressive Conditional Heteroscedasticity (ARCH) model. We define the model by a hierarchically: first define $X_t = \sigma_t Z_t$ where $Z_t \sim N(0,1)$ i.i.d., but treat σ as being random such that

$$\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \cdots + \alpha_q X_{t-q}^2$$

So the variance is time dependent – a large value of X_t will result in period of high volatility. We illustrate an example of a realisation in Fig. 1.12

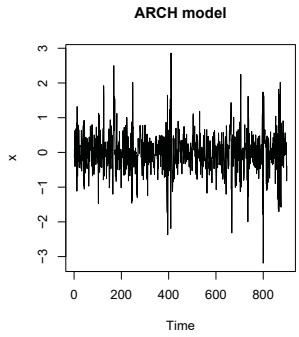

Figure 1.12: Realisation of ARCH process

1.3 Computing in R

In R the basic object is a time-series object. The function `ts( )` is used to create such an object. For example for the data in Example 1.2.6 we can use the functions

```
> gas <- ts(gas.raw, start = 1960, frequency = 12)
```

The following commands can be used on the `ts`-objects.

```
> plot(gas, main="Ontario gas prices")
> start(gas)
> end(gas)
> frequency(gas)
```

1.4 Forecasting, prediction and control problems

The following are some examples of forecasting and control problems with a discussion of the objectives of the problem and the information that is available to be used in the forecast.

Example 1.4.1. (San Diego house prices) In Chapter 8 of Gonzalez-Rivera (2013) we find an analysis of forecasting the San Diego house quarterly price index. This is a summary statistic for the property market in San Diego, shown in Fig. 1.13(a) and the percentage change in price (Panel (b)).

We see that the index itself looks non-stationary but there is some stability in growth rate. Using one of the models proposed by Gonzalez-Rivera (2013) we show a point estimate of the growth rate (red) and prediction range (blue), the dash horizontal line is the sample mean.

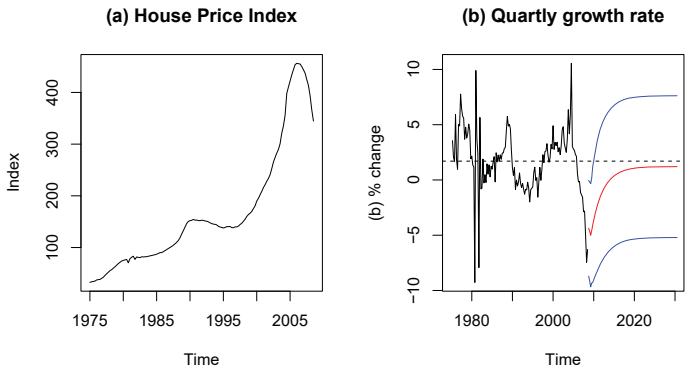

Figure 1.13: San Diego housing market

Gonzalez-Rivera (2013, Page 215) identifies four different agents who might be interested in forecasts here: Property owners: a large amount of

householder's wealth is the value of the house, so managing wealth requires understanding what might happen to this value in the future. Real estate agents; if capital gains are likely in a given market then investors may decide to invest. Government: Policy makers need to think about the effect on the house market of, say interest rates. Mortgage banks: deciding on rules for giving potential buyers a mortgage depends on understanding the possible future behaviour of the market in the time period of the mortgage.

She goes on to note that different agents may have different loss functions thus even with the same model we make have different forecasts. For example the loss associated with overestimating the index may have, for some agents, a more sever effect than underestimating it, so asymmetric loss functions may be appropriate, see Chapter 2. We can also think about the length of time that different agents may be interested in forecasting. Someone thinking about selling a house in the next year would have four quarters as their forecast window, where as a real-estate company may be interested in a much longer time scale. The plot Fig. 1.13(b) is qualitatively typical of many forecasts. Short term forecasts – here a few quarters – are close to the current values, but as the forecast window lengthens the forecast converges to the (unconditional or marginal) mean of the sample. The variance of the forecast is smaller for short term forecast and grow until it convergences to (unconditional or marginal) variance of the sample. For this to be valid we have to assume stationarity in the data, see Chapter 3 for details. In particular that the marginal mean and variance in the observed data are a good representative for longer time periods – compare to Example 1.2.13 where the stationary distribution can is used for long term prediction. In the San Diego housing market example the stationarity assumption becomes more questionable the longer the time period of the forecast that is being considered.

In Example 1.4.1 we are using the information in the housing index itself as the sole source of information in order to make our forecasts. We might, very reasonably ask the question as to why that is a reasonable thing to do? Are the not other information sources that could be used? A possible justification might come from the efficient market hypothesis.

Definition 1.4.2. (Efficient market hypothesis) The hypothesis states that in an efficient market the price of an object always incorporates and reflects all relevant information.

We, of course, do not know if the San Diego housing market is efficient which means that we could, in principle look for other variables with predictive power.

Example 1.4.3. (Economic forecast of GDP) We often hear about forecasts of the growth of the gross domestic product (GDP) of a country. This is

an index which measures the size of an economy. In fact at any given time there would be uncertainty not just about future values of GDP but also the current value. It takes time for all the inputs to GDP to be collected and estimates of its value given in one month are often revised later.

There are economic variables which have predictive power for forecasting changes in GDP in the next few quarters but are available for use at the current time. These are called *leading indicators* [2] These include: the stock prices of 500 common stocks, interest rate spread, manufacturers' new orders and weekly claims for unemployment insurance. These leading indictors can be used are covariates in an econometric model – often a regression model – which can then be used to make short term forecasts with associated measures of uncertainty. The issue of how to build a good predictive regression model is looked at in Chapter 2.

In this example, as with the house price example there may be a number of different agents who would be interested in a forecast of GDP. For example the central bank of a country and a multinational company looking to invest in a new market. The use of the forecast is different between these two agents. The central bank may be interested in the forecast as part of a *control problem*. They have some control of interest rates which are drivers of GDP. There will be a feedback between the level of interest rates and the forecast of GDP. The multinational company could use the forecast to make a binary decision about whether to invest or not in a new market.

The regression models mentioned in the previous example are a very powerful tool for forecasting and prediction. In these notes we will tend to use the term forecasting when there is an underlying time series structure and prediction otherwise.

Example 1.4.4. (Prostate cancer example) (Hastie et al., 2009, Chapter 3) describes an example, based on a study by Stamey et al. (1989). Here the prediction problem is to estimate the level of a clinically important, but hard to measure directly, antigen value, using a number of variables which are easier to measure.

The problem can be thought of as a regression problem with the antigen being the response and the easy to measure variables possible explanatory variables. There are in fact many variable which have some correlation with the response. These explanatory variables are often correlated with each other – i.e. share much of the same information. We know from regression courses that the best prediction model is not the one that contains all possible explanatory variables. Rather we need to balance parsimony (the simplicity of the model) with goodness-of-fit. In Chapter 2 we investigate how to perform the model building in examples such as this and how to understand the predictive power of regression models.

[2] https://www.conference-board.org/data/bcicountry.cfm?cid=1

The use of the forecast of DGP by a central bank in Example 1.4.3 was part of a control problem i.e. deciding what value to set interest rates. There is a strong relationship between statistical forecasting and control problems.

Example 1.4.5. (Spacecraft control) Forecasting questions are often part of a larger problem of how to control complex and noisy systems using feedforward and feedback loops. An example of this would be a navigation system on a robotic spacecraft. The probe needs to be able to control its navigation using sensor data – which can be noisy – in a time sensitive way. The forecast comes in answering the question: what is the future position of the craft if current settings of the controls are kept stable? and the related question of: how should be change the controls to ensure that the position of the spacecraft stays on target?

State space methods, as explored in Chapter 5, including the Kalman Filter are a powerful tool in such control problems. The filter is a recursive method which uses statistical models to combine new measurements from the sensors relative to past information. It also determines up-to-date uncertainties of the estimates for real-time quality assessments.

We have seen that in order to make forecasts we need to understand – and model – different forms of non-stationary. Example 1.1, 1.2.6 and 1.2.8 all showed examples where periodic changes, on possibly different scales, where part of the non-stationary. With complex systems it is not always clear if there is periodic behaviour or what the period is. We can get insight into the periodic structure by representing the time series in the frequency domain.

Example 1.4.6. (Sunspot data) Sunspots lie on the photosphere of the Sun that appear as dark spots. The left hand panel of Fig. 1.14 shows the sunspot numbers from 1749 to 1983 which was collected at Swiss Federal Observatory, Zurich until 1960, then Tokyo Astronomical Observatory and can be found in R. We see a complex pattern of periodic behaviour. The spectral decomposition of a time series decomposes the series into a sum of sinusoidal components with different frequencies. The righthand plot shows the size of these components by the frequency and we see a peak at between $2\pi/11$ and $2\pi/10$ which corresponds to an approximate cycle with period around 10 to 11 years.

Since this is a statistics course we focus our attention on methods which are fundamentally statistical – where information about the forecast is extracted from observed data, maybe through the use of a model. For completeness we note that other methods are often used in practice.

Definition 1.4.7. (Non-statistical forecasting methods) *Scenario analysis* looks at the forecasting by considering a (small) finite number of alternative possible outcomes, by asking the question: 'what if?' In general the

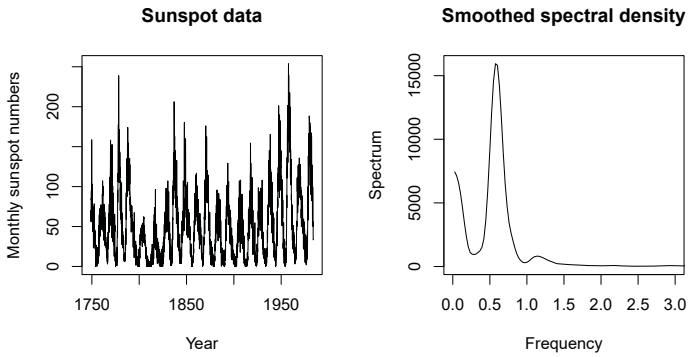

Figure 1.14: Sunspot data: time domain and frequency domain representations

probability of events are not considered, see for example Van der Heijden (2011). The *Delphi method*, define in (Linstone et al., 1975, p. 3), as 'a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem.' It is used when analytical techniques are not available and subjective judgments need to be used. These judgements though come from groups of individuals with diverse backgrounds with respect to experience or expertise.

1.5 Simple statistical tools

In this section we look at some simple statistical tools which can be used with the time series such as those in Section 1.2. These typically use minimal modelling assumptions and are often used for exploratory or descriptive purposes.

Definition 1.5.1. (Model with trend) A trend model for the time series X_t is a decomposition

$$X_t = m_t + Y_t,$$

where m_t is a slowly varying function and Y_t has zero mean.

Note that in Definition 1.5.1 we have not given a formal definition to *slowly varying* but in practice we need that the change in the function is slow enough that we get a good estimate using the observed data. Thus examples could be

$$m_t = \alpha_0 + \alpha_1 t,$$

or

$$m_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2.$$

Here there are only a few parameters which are needed to be estimated so with a reasonable sample size we might make estimates which are good enough for purpose. Note that Definition 1.5.1 is weaker than a regression models since we are not assuming anything about dependence structure of Y_t nor its variance.

Definition 1.5.2. (Model with seasonal component) A model with a seasonal component with period d for X_t is a decomposition

$$X_t = s_t + Y_t,$$

where s_t satisfies $s_t = s_{t+d}$, for all t.

Simple examples would be monthly data with $d = 12$, weekly data with $d = 52$ etc. We can put these two together to get a linear decomposition model

Definition 1.5.3. (Linear decomposition model) A linear decomposition model for the time series X_t is a decomposition

$$X_t = m_t + s_t + Y_t,$$

where $E(Y_t) = 0$, m_t is a slowly varying function, s_t is periodic with period d and, for identification reasons we further assume

$$\sum_{t=1}^{d} s_t = 0.$$

We can construct a *multiplicative decomposition* model by applying a linear decomposition to $\log X_t$.

Example (1.2.6 revisited). In R we can estimate the components of the linear decomposition model by using the function `decompose(x, type = c("additive", "multiplicative"))`. We see the result of this on the Ontario gas data in Fig. 1.15. We see the time series plot in the top panel, an estimate of the trend m_t in the second panel, an estimate of the seasonal effect, s_t,in the third panel and the 'residual' random term in the lower panel. We note that these models do not assume the usual regression model conditions on the random term – in particular we see that we do not have constant variance and we say nothing about the dependence structure.

Example (1.2.11 revisited). In R we can estimate the components of the linear decomposition model by using the function `decompose(x, type = c("additive", "multiplicative"))` so, for example we get Fig. 1.16 by using

```
> plot(decompose(birth, type="additive"))
```

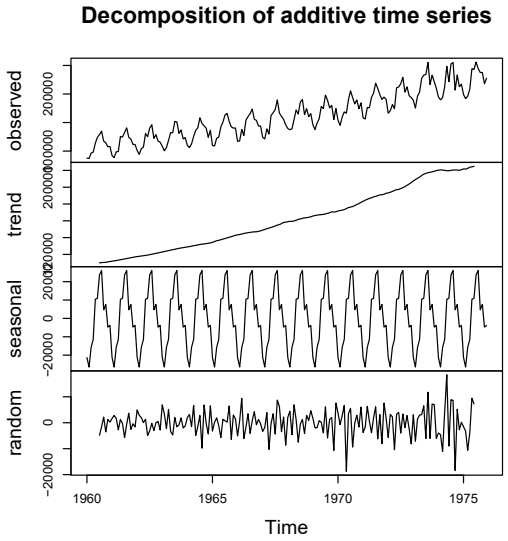

Figure 1.15: Ontario gas demand (gallons)

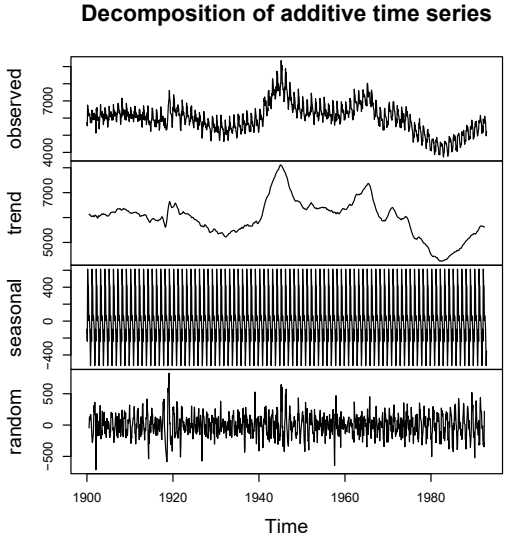

Figure 1.16: Monthly birth data in Denmark 1900-1992

Definition 1.5.4. (Simple moving average filter) For Model (1.5.3) we can estimate with m_t with a *moving average filter*. Assume that the period is

even $d = 2q$, then the filter is defined on the realisation x_t by

$$\widehat{m}_k = \frac{0.5x_{t-q} + x_{t-q+1} + \cdots + x_{t+q-1} + 0.5x_{t+q}}{q}.$$

Since we have that $\sum_{i=1}^{d} s_t = 0$. the seasonal component will not be part of this estimate.

We can then estimate the seasonal components. For each $k = 1, \ldots, d$ compute the average w_k of

$$\{x_{k+jd} - \widehat{m}_{k+jd} | q < k + jd \leq n - q\}.$$

We then normalise to get

$$\widehat{s}_k = w_k - \frac{\sum_1^d w_j}{d}.$$

That is for monthly data we average over all January residuals, all February residuals etc.

Example (1.2.11 revisited). If we look at the Danish birth data we can estimate the level, m_t, using a 12 point moving average filter. Figure 1.17 shows both the raw data (black) and the estimated level $\widehat{m}_t$ which smooths out the seasonal component (red).

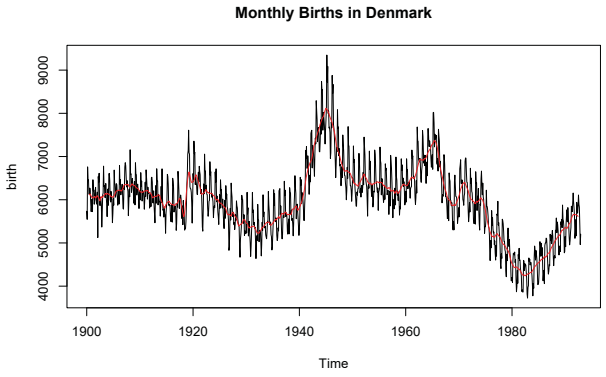

Figure 1.17: Danish birth data: black raw data and red estimated level

We note that the change point around 1919 has not be well estimate by the filter. Rather than having a discontinuity we have a estimated a linear slope in the year containing the discontinuity. The key point here is that smoothing methods can *over-smooth* real features in the data. This can also be seen in the random component where there is an increase in the variance around the discontinuity. We also see other regions where the variance is inflated.

The moving average filter of Definition 1.5.4 is not the only way of filtering the data. One disadvantage of it symmetric form is that it can only estimate the m_t in the middle of the data – as can be seen from Fig. 1.17. This means it can't be used directly for forecasting. There are other filtering methods which over come this. We look at a useful methods and related methods, exponential smoothing and, the more general, Holt-Winters filtering.

Definition 1.5.5. (Exponential smoothing) Exponential smoothing can be used for a time series to estimates the level of the process m_t which is assumed slowing varying. It should not be used when there is a trend or seasonality. Assume that we can observe $x_1, \ldots, x_n$. We select an initial value m_0, typically x_1 then we update the estimate recursively via the update equation

$$m_{t+1} = \alpha x_t + (1-\alpha)m_t = m_t + \alpha(x_{t+1} - m_t)$$

The second from of the update is called the error correcting version and

$$e_{t+1} := x_{t+1} - m_t$$

is the one-step ahead forecast error. The tuning parameter α needs to be selected and this can be done by finding the α which minimises

$$\sum_{t=1}^{n} (x_{t+1} - m_t)^2.$$

Definition 1.5.6. (Holt-Winters filtering) The Holt-Winters method generalises exponential smoothing to the case where there is a trend and seasonality. We have three terms which depend on time t. The first is a *level* a_t, the second is the *trend* b_t and the third is the *seasonal component* s_t. The terms a_t, b_t are considered slowly varying and the mean of the time series at time $t+h$ is given by

$$m_{t+h} = a_t + b_t h + s_{t+h}$$

The Holt-Winters prediction function for h time periods ahead of current time t is $\widehat{x}_{t+h} = a_t + b_t h + s_{t+h}$. The *error term* for this prediction is then defined as

$$e_t := x_t - (a_{t-1} + b_{t-1} + s_{t-p})$$

then starting with initial values for a, b and s and then update the parameters according to the size of the error over the range of t by using

$$\begin{aligned} a_t &= a_{t-1} + b_{t-1} + \alpha e_t, \\ b_t &= b_{t-1} + \alpha\beta e_t, \\ s_t &= s_{t-p} + \gamma e_t. \end{aligned}$$

The method uses three tuning parameters α, β and γ and these are selected by minimising the sum of squared errors.

Example (1.2.11 revisited). In R we can apply the Holt-Winters method on the Denmark data and predicting using the result with the functions

```
> birth.hw <- HoltWinters(birth)
> birth.hw.predict <- predict(birth.hw, n.ahead=12*8)
```

and we can see the fit (red) and the forecast (blue) for the next 8 years in Fig. 1.18. The prediction models the seasonality and the prediction is based on a local linear approach. This is a strength for short term forecasting but, as the figure shows, may be unrealistic for longer term forecasts.

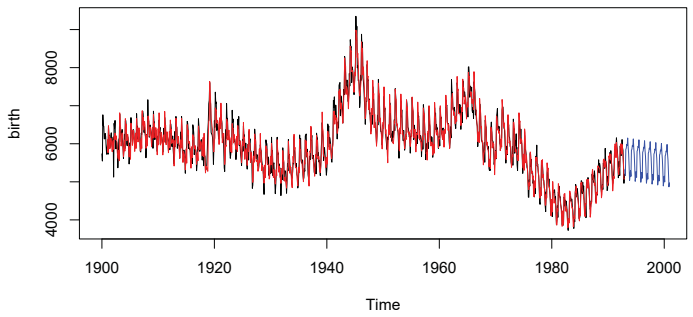

Figure 1.18: Monthly birth data in Denmark 1900-1992: fitted red and forecast blue

2

Regression methods and model building principles

2.1 Introduction

Much of this chapter follows the approach of Hastie et al. (2009, Ch. 1, 3 and 7) and there is a great deal of very interesting further reading in that book. We focus here mostly on regression models but the ideas of this chapter have a much wider applicability in statistical classification and machine learning. The course STAT 441 Statistical Learning - Classification develops many of these themes.

2.2 Statistical decision theory

Let $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$ be vector valued random variables where we are trying to predict Y given the information available in X. We are therefore looking for a function $f : \mathbb{R}^p \to \mathbb{R}$, which has 'nice' properties.

Definition 2.2.1. (Loss function) A loss function $L(Y, f(X))$ is a real valued function which measures the error in estimating Y with the estimate $f(X)$. Typically we assume $L(y, f(x)) \geq 0$ with equality if and only if $y = f(x)$. Since $L(Y, f(X))$ is a random variable, we may also consider its expected value, $E_{X,Y}\left[L(Y, f(X))\right]$, often called the *risk*. Here the expectation is over the joint distribution of X, Y.

Example 2.2.2. (Quadratic loss) A commonly used example is the squared error loss, or *quadratic*, loss

$$L_{SE}(Y, f(X)) = (Y - f(X))^2.$$

The expected value is called the mean squared error (MSE)

$$MSE(f) := E_{X,Y}\left(L_{SE}(Y, f(X))\right) = E_{X,Y}\left(\left(Y - f(X)\right)^2\right). \tag{2.1}$$

Theorem 2.2.3. (Conditional expectation) If X and Y are two random variables with $E(Y) = \mu$ and $Var(Y) < \infty$, then the function, f, which minimises $MSE(f)$, for X, Y, is given by the conditional expectation.

$$f(X) = E_{Y|X}(Y|X).$$

Proof. (Outline)

1. Show that the constant c that minimises $E[(Y-c)^2]$ is $c = \mu$.
2. Show that the function $f(\cdot)$ that minimises $E[(Y - f(X))^2|X = x]$ is $f(x) = E(Y|X = x)$.
3. If X, Y have a joint density function $f_{X,Y}(x, y)$ write down the $E[(Y - f(X))^2]$ and $E[(Y - f(X))^2|X = x]$ in terms of integrals.
4. Show that if Z is a positive random variable and its expectation exists then $E(Z) \geq 0$
5. Deduced that the random variable $f(X)$ that minimises $E[(Y-f(X))^2]$ is $f(X) = E(Y|X)$

□

Theorem 2.2.4. (Absolute loss) We can also define a loss function in terms of absolute values, i.e.

$$L_{abs}(Y, f(X)) := |Y - f(X)|$$

the function which minimises $E(L_{abs}(Y, f(X)))$ is

$$\widehat{f}(x) = Median(Y|X = x).$$

Proof. (Outline) Similar to Theorem 2.2.3. If $f_Y(y)$ is the density of Y, we write

$$E[L_{abs}(Y, f(c))] = \int_{-\infty}^{\infty} |y-c| f_Y(y)dy = \int_{-\infty}^{c} (c-y) f_Y(y)dy + \int_{c}^{\infty} (y-c) f_Y(y)dy$$

Differentiating with respect to c and setting to zero gives turing point $\widehat{c}$ satisfies $F(\widehat{c}) = 0.5$, i.e. the median is the turning point. □

Definition 2.2.5. (Zero-one loss) If Y is a discrete, or categorical random variable, with sample space $\mathcal{S}$, then a commonly used loss function is the indicator, or the zero-one, function

$$L_{01}(Y, f(X)) = \begin{cases} 0 & \text{if } Y = f(X), \\ 1 & \text{if } Y \neq f(X). \end{cases}$$

and then its expected value is

$$E(L_{01}(Y, f(X))) = P(Y = f(X)).$$

Theorem 2.2.6. (Bayes classifier) The function which minimises $E(L_{01}(Y, f(X)))$ for X, Y where Y has sample space $\mathcal{S}$ is given by

$$\widehat{f}(x) = \max_{y \in \mathcal{S}} P(y|X = x),$$

the so-called *Bayes classifier*.

2.3 Linear regression models

Let us start by considering an example where linear regression is often used for prediction and the quadratic loss function is a natural one.

Example 2.3.1. (House price example) The market value of a house is of interest to a both buyers and sellers. It should be a function of a number of features of the house and a multiple linear regression model can be used to estimate this function. In order to calibrate the model a data set has been constructed using the following variables.

X_1	Current taxes
X_2	Number of Bathrooms
X_3	Lot size
X_4	Living space
X_5	Number of parking spaces
X_6	Number of rooms
X_7	Number of bedrooms
X_8	Age of house
X_9	Number of fireplaces
Y	Actual sale price

From Theorem 2.2.3 we see that the best forecast of Y given the information in X is given by $E(Y|X)$ in terms of MSE. Computing a conditional expectation requires some knowledge of the joint distribution. The linear regression model makes some simplifying assumptions, firstly if $X = (X_1, \ldots, X_p)$ then we assume

$$f(X; \boldsymbol{\beta}) = E(Y|X) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i \tag{2.2}$$

In detail we have by defining the design matrix $\mathbf{X}$ and response and error vectors $\mathbf{y}$ & $\boldsymbol{\epsilon}$, respectively

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} = \begin{bmatrix} 1 & x_{1\,1} & x_{1\,2} & \ldots & x_{1\,p} \\ 1 & x_{2\,1} & x_{2\,2} & \ldots & x_{2\,p} \\ & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n\,1} & x_{n\,1} & \ldots & x_{n\,p} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \text{ and } \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

we write $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\epsilon}$.

Definition 2.3.2. (Residual sum of squares) To estimate $\boldsymbol{\beta}$ given a training set $\{(y_i, \boldsymbol{x}_i)|i = 1, \cdots, N\}$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ it is common to minimise the residual sum of squares

$$RSS(\boldsymbol{\beta}) := \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

The ordinary least squares estimate $\widehat{\boldsymbol{\beta}}$ is defined as $\arg\min_\beta RSS(\boldsymbol{\beta})$.

For notational convenience define $\boldsymbol{X}$ to be the $N \times (p+1)$ matrix where each row corresponds to one of the elements in the training set, and we define $x_{i0} = 1$ for all i, and $\boldsymbol{y} = (y_1, \ldots, y_N)^T$. With this notation we have

$$RSS(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta)\,. \tag{2.3}$$

Theorem 2.3.3. (OLS estimation) The ordinary least squares estimate is given by

$$\widehat{\boldsymbol{\beta}}_{OLS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

when $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ exists.

Proof. Using matrix differentiation rules, see Appendix 2.6.1, we get

$$\frac{\partial}{\partial \boldsymbol{\beta}} RSS(\boldsymbol{\beta}) = -2\boldsymbol{X}^T (y - \boldsymbol{X}\boldsymbol{\beta}) \tag{2.4}$$

and

$$\frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^T} RSS(\boldsymbol{\beta}) = -2\boldsymbol{X}^T\boldsymbol{X}$$

Setting Equation 2.4 to zero and solving for $\boldsymbol{\beta}$ gives the result. □

Definition 2.3.4. (Hat matrix) We define the, so-called, hat matrix as

$$\boldsymbol{H} := \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T,$$

which is an orthogonal projection matrix since

$$\boldsymbol{H}^2 = \boldsymbol{H},$$

The *fitted values* are the projection of the observed values in the training set $\boldsymbol{y}$ projected onto the linear subspace spanned by the columns of $\boldsymbol{X}$, i.e.

$$\widehat{\boldsymbol{y}} := \boldsymbol{H}y.$$

The estimated residuals are the corresponding orthogonal complement

$$\widehat{\boldsymbol{y}} := \boldsymbol{y} - \boldsymbol{H}\boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{H})\,\boldsymbol{y}.$$

Theorem 2.3.5. (Properties of hat matrix)

(a) The hat matrix $\mathbf{H}$ is $n \times n$ *symmetric*, i.e. $\mathbf{H} = \mathbf{H}^T$,

(b) The hat matrix $\mathbf{H}$ is *idempotent*, i.e. $\mathbf{H}^2 = \mathbf{H}$,

(c) $\widehat{\mathbf{y}} = \mathbf{H}\mathbf{y} = \mathbf{X}\,(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ is the projection of the vector $\mathbf{y}$ onto the column space of the design matrix $\mathbf{X}$.

(d) $\text{trace}(\mathbf{H}) = \sum_{i=1}^{n} h_{i\,i} = p + 1$.

(e) For a given point $\mathbf{x}$, we can write $\mathbf{h}(\mathbf{x})^T = \mathbf{x}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}$ and therefore

$$\widehat{f}_n(\mathbf{x}) = f(\mathbf{x};\,\widehat{\boldsymbol{\beta}}) = \mathbf{h}(\mathbf{x})^T Y_i = \sum_{i=1}^{n} h_i(\mathbf{x})\,Y_i\,.$$

Proof. Sketch of (d): Recall the trace of a square matrix being the sum of its diagonal elements, then note that

$$\begin{aligned}\text{trace}(\mathbf{H}) &= \text{trace}(\mathbf{X}\,(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T) \\ &= \text{trace}((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\,) \\ &= \text{trace}(\mathbf{I}_{p+1}) = p + 1\,.\end{aligned}$$

□

We shall now assume that we treat the response variables $\boldsymbol{X}$ as fixed – or alternatively think of conditioning on the observed values of the explanatory variates. Further, assume that $Var(Y_i) = \sigma^2$ and that Y_i and Y_j are uncorrelated.

Theorem 2.3.6. (Variance of OLS) The variance-covariance of the sampling distribution of $\widehat{\boldsymbol{\beta}}_{OLS}$ is given by

$$Var\left(\widehat{\boldsymbol{\beta}}_{OLS}\right) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\sigma^2$$

Proof. From Theorem 2.3.3 we have that $\widehat{\boldsymbol{\beta}}_{OLS}$ is a linear function of $\boldsymbol{y}$ which has a variance-covariance matrix

$$Var(\boldsymbol{y}) = \sigma^2\boldsymbol{I}_{N\times N},$$

hence we have the result from standard properties of covariances. □

It can be helpful to think of the variability in the data in terms of different types of 'sums-of-squares'.

Definition 2.3.7. (Sum of squares)

(a) Total corrected sum of squares

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \overline{y})^2 .$$

(b) Regression sum of squares (i.e., sum of squares explained by the regression model),

$$\text{SSReg} = \sum_{i=1}^{n} (\widehat{y}_i - \overline{y})^2 .$$

(c) Residual sum of squares

$$\text{RSS} = \text{RSS}(\widehat{\beta}) = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2 .$$

It is easy to show that TSS = SSReg + RSS, i.e.

$$\begin{array}{ccccc} \text{total variability} & = & \text{variability explained} & + & \text{unexplained variability} \\ & & \text{by the model} & & \text{(i.e. error)} \end{array}$$

Definition 2.3.8. (Unbiased estimate of variance) The unbiased estimate of σ^2 is defined as

$$\widehat{\sigma}^2 := \frac{1}{N-p-1} \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2.$$

Theorem 2.3.9. (Sampling distributions) As well as the assumptions of Theorem 2.3.6, further assume that we can write $\boldsymbol{y}$ in terms of its mean and an error term

$$\begin{aligned} \boldsymbol{Y} &= E(\boldsymbol{Y}|\boldsymbol{X}) + \boldsymbol{\epsilon} \\ &= \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon. \end{aligned}$$

where $\boldsymbol{\epsilon} \sim N\left(0, \sigma^2 \boldsymbol{I}_{N\times N}\right)$.

(i) The sampling distribution of $\widehat{\boldsymbol{\beta}}_{OLS}$ is given by

$$\widehat{\boldsymbol{\beta}}_{OLS} \sim N\left(\beta, \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}\right).$$

(ii) The distribution of $\widehat{\sigma}^2$ is determined by

$$(N-p-1)\frac{\widehat{\sigma}^2}{\sigma^2} \sim \chi^2_{N-p-1}.$$

(iii) To test the hypothesis $\beta_j = 0$ we use the test statistic

$$z_j = \frac{\widehat{\beta}_j}{\widehat{\sigma}\sqrt{\nu_j}}$$

where ν_j is the j^{th} diagonal element of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$, and z_j has a t_{N-p-1} distribution under the null.

(iv) To test a smaller model M_0, with p_0 covariates, inside a larger model M_1, with p_1 covariates, we use the test statistic

$$F = \frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N - p_1 - 1)}$$

where RSS_i are the residual sum of squares for each estimated model. F has an $F_{p_1-p_0,N-p_1-1}$ distribution under the null.

For more details of the tools available with regression see Appendix 2.6.2.

Example (2.3.1 revisited). In R we can fit the full model using OLS and get the following output.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.31044    5.96093   2.568   0.0223 *
X1           1.95413    1.03833   1.882   0.0808 .
X2           6.84552    4.33529   1.579   0.1367
X3           0.13761    0.49436   0.278   0.7848
X4           2.78143    4.39482   0.633   0.5370
X5           2.05076    1.38457   1.481   0.1607
X6          -0.55590    2.39791  -0.232   0.8200
X7          -1.24516    3.42293  -0.364   0.7215
X8          -0.03800    0.06726  -0.565   0.5810
X9           1.70446    1.95317   0.873   0.3976
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 2.973 on 14 degrees of freedom
Multiple R-squared:  0.8512,Adjusted R-squared:  0.7555
F-statistic: 8.898 on 9 and 14 DF,  p-value: 0.0002015
```

We see that most of the terms individually are 'not significant' but the whole model has an $R^2 = 0.8512$, and an overall p-value of 0.0002015 so appears to explain a good deal of the variation of Y and hence might be a useful predictive model.

Individually the estimated β_i values are a bit puzzling, for example the coefficient of X_6, the number of rooms, is negative: intuitively we might feel that the more rooms a house has the higher the price. Of course the estimate is not 'statistically significantly negative' and maybe it could be set to zero. We return to this in the following section.

In Theorems 2.3.3 and 2.3.9 to both define the OLS estimates and understand their sampling properties we need the condition that n $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ exists, i.e., $(\boldsymbol{X}^T\boldsymbol{X})$ is non-singular. One or the most hardest parts of using regression models is the case where $(\boldsymbol{X}^T\boldsymbol{X})$ is close to being singular.

Definition 2.3.10. (Multicollinearity) If two, or more, of the columns of $\boldsymbol{X}$ are highly correlated we say that we have multicollinearity.

In in terms of prediction we have that the correlated explanatory variables are essentially are sharing the same information about Y. Having highly correlated explanatory variables does not add much to predictive power but is statistically expensive since there are more parameters to be estimated. We see in later sections that the bias is not reduced by much but the variance is increased, see §2.4.1. In particular the term ν_j in Theorem 2.3.9 can very large meaning we can not make very precise statements about the corresponding parameter.

Example (2.3.1 revisited). Going back to the house price example we have of 9 explanatory variates and the intercept term. Figure 2.6 shows a pairwise scatterplot of the 9 explanatory variables and we see that there are correlations. For example X_3, *lot size* and X_4 *living space* are correlated – as might be expected – and so share some of the same predictive power.

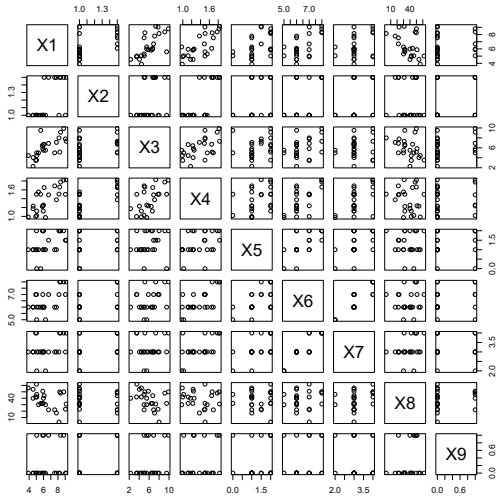

Figure 2.1: Pairwise scatterplot for covariate in house example

In the case of multiple linear regression suppose we want to predict the value of Y_0 for a new explanatory vector $\boldsymbol{X_0}$ when we have a training set $\mathcal{T}$ available from which we can estimate $\widehat{\beta} = \widehat{\beta}(\mathcal{T})$.

Theorem 2.3.11. (Prediction interval in regression) If we assume that $(Y_0, \boldsymbol{X_0})$ is independent of the training set $\mathcal{T} = \{(Y_1, \boldsymbol{X_1}), \ldots, (Y_N, \boldsymbol{X_N})\}$ and assume we have a correctly specified model.

Define the point forecast as

$$\widehat{\mu} = \boldsymbol{x_0}^T \widehat{\beta}$$

i.e. the estimated value of the conditional expectation. Then a α%-prediction interval is

$$\widehat{\mu} \pm c_\alpha \widehat{\sigma} \sqrt{1 + \boldsymbol{x_0}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x_0}}$$

where c_α is the $1 - \frac{\alpha}{2}$-quantile from the t_{N-p-1}-distribution and $\widehat{\sigma}^2$ is defined by Definition 2.3.8.

Proof. (Sketch) By independence of the prediction and training set, the distribution of Y_0 and the distribution of $\widehat{\beta}$ we can show that

$$Y_0 - \boldsymbol{x_0}^T \widehat{\beta} \sim N\left(0, \sigma^2 \left(1 + \boldsymbol{x_0}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x_0}\right)\right).$$

Since σ^2 has to be estimated we define

$$T := \frac{Y_0 - \boldsymbol{x_0}^T \widehat{\beta}}{\widehat{\sigma} \sqrt{(1 + \boldsymbol{x_0}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x_0})}}$$

which has a t_{N-p-1} distribution, when $\widehat{\sigma}$ is the sample estimate of σ.

This means, for all parameter values we have

$$P(-c_\alpha \leq T \leq c_\alpha) = 1 - \frac{\alpha}{2}.$$

or by rearranging have a $1 - \frac{\alpha}{2}$ probability that Y_0 lies in the interval

$$\left(\boldsymbol{x_0}^T \widehat{\beta} \pm c_\alpha \widehat{\sigma} \sqrt{(1 + \boldsymbol{x_0}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{x_0})}\right)$$

□

Example 2.3.12. (Prediction in regression) The following simple simulated example shows both how forecasting in prediction works and illustrates a major weakness of this, and other forecasting methods. Consider Fig. 2.2, panel (a) shows the data (black dots) a fitted linear model (solid red line) and prediction intervals (solid blue) for 'new' values of the covariate of $6, 12$ and 18.

Panels (b) and (c) show information about the residuals and it looks like the linear model fits very well. In fact, the true model here was non-linear (dashed red line) and the prediction interval for $x = 12$ and 16 would be very poor.

The lesson to learn from Example 2.3.12 is that making a prediction where the new covariate values are far from the values used to fit the data is a very risky thing to do. The goodness-of-fit statistics only tells us that the model fits the training data and very little about how the model behaves when far from the training data.

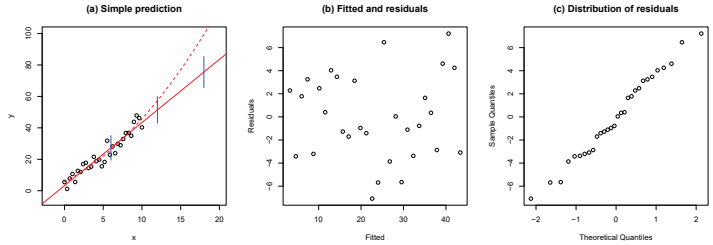

Figure 2.2: Prediction with regression

2.3.1 Computing in R

In R the key function for regression is the `lm( )`. For example in Example 2.3.1 we fit the model using

```
>  house.fit <- lm(Y ~X1+X2+X3+X4+X5+X6+X7+X8+X9, data=house.price)
```

the `ls`-object then contains all the output needed for analysis. Functions for which it is an argument includes the following:

```
> summary(house.fit)

> plot(house.fit)

> house.fit$fitted

> house.fit$residuals
```

The last two of these are very helpful for looking at good-of-fit residual analysis plots. When we want to use the `lm( )` function for prediction we can use the `predict( )` or `predict.lm( )` function. For example, suppose we look a the simple model which regresses house price Y against $X1$ and $X2$. We fit the model using

```
 house.fit1 <- lm(Y ~ X1+X2, data=house.price)


> new <- data.frame(X1=c(6.00), X2= c(1.5) )
> predict.lm(house.fit1, new, interval="prediction")
       fit      lwr      upr
1 35.56195 29.05033 42.07357
```

Here `fit` is our point estimate of the price and `lwr` and `upr` define a 95%-prediction interval thought the lower and upper values respectively.

2.4 Model complexity

The theory of linear models in the previous section, for example Theorems 2.3.3 and 2.3.9, implicitly assumes that we know the form of the regression model. That is Equation (2.2) holds for a known set of p explanatory variables. We can use methods based on residuals for model checking – for example Fig. 2.2(b, c) – but there can be many models which have 'good enough' goodness-of-fit to be considered plausible. Which one to use? We shall see that we do not use the model which has the best fit.

2.4.1 Bias-variance decomposition

Let us consider the multiple regression model with a large number of covariates and how the fitted model might be used for prediction. Naively, one might think that the more covariates there are the more information is available, so the better the forecast. Multicollinearity, Def. 2.3.10, is one reason for that this is not true. Adding covariates which are strongly correlated with ones already included typically reduces the model's prediction power.

There are, in fact, even stronger reasons why over-complex models should be avoided. This is called *over-fitting*, and this is where the fitted model is a very good description of the training data – the sample – but not such a good description of the underlying population. Over-fitted models have poor power to generalised away from the training data.

Definition 2.4.1. (Bias-variance decomposition) Let $\mathcal{T}$ be the training set used to estimate $\widehat{\beta} = \widehat{\beta}(\mathcal{T})$, we now want to estimate the forecasting error for a new set of covariates, x_0. Let the forecast be

$$\widehat{y}_0 := x_0^T \widehat{\beta}$$

If y_0 is the actual value we are trying to forecast, and using quadratic loss (Example 2.2.2) we evaluate the forecast using

$$\begin{aligned} MSE(x_0) &:= E_{\mathcal{T}}\left(\{y_0 - \widehat{y}_0\}^2\right) \\ &= E_{\mathcal{T}}\left(\{y_0 - E_{\mathcal{T}}(\widehat{y}_0) + E_{\mathcal{T}}(\widehat{y}_0) - \widehat{y}_0\}^2\right) \\ &= (y_0 - E_{\mathcal{T}}(\widehat{y}_0))^2 + E_{\mathcal{T}}\left(\widehat{y}_0 - E_{\mathcal{T}}(\widehat{y}_0))^2\right) \\ &= Bias^2(\widehat{y}_0) + Var_{\mathcal{T}}(\widehat{y}_0). \end{aligned}$$

So when we are evaluating the MSE of a prediction we typically are trying to balance two different objectives: the bias of the forecast and the variance. In previous statistics courses we have often chosen estimates with zero bias, for example Def. 2.3.8 defines a *unbiased* estimate of variance and Theorem 2.3.9 shows that the OLS estimate of β is unbiased for a correctly specified

model. However Hastie et al. (2009, §3.3) criticises the OLS method for models with many covariance for prediction since although the predictions have small bias they can have large variance. Prediction accuracy can be improved by shrinking or setting some of the coefficients of $\widehat{\beta}$ to zero. This increases bias – since we are changing an unbiased estimate – but reduces variability.

Figure 2.3 is an attempt to give a geometric view of the trade-off between bias and variance. Suppose we chose a parameterised model space – say linear regression with independent normal errors and a fixed set of explanatory variates – the 'true' model probably does not lie in the space but hopefully is close to it. We can think of the error caused by the true model being outside the space as model error or bias. Since the model spaces involves a set of unknown parameters their estimation involves uncertainty, which is variability. We can reduce the bias by making the model space more complex – say by adding new covariates or allowing dependence in the model – this will mean there are more parameters to estimate, so more variability.

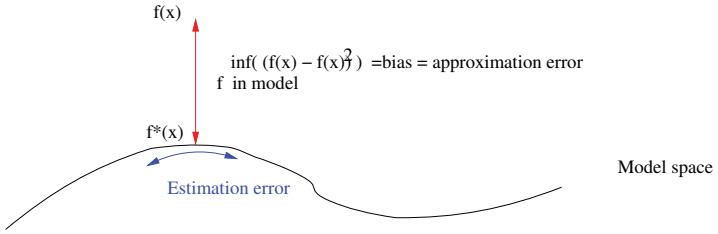

Figure 2.3: Bias Variance trade off

2.4.2 Subset selection

One approach to improving the predictive performance is to use only a subset of the explanatory variables. This may reduce the multi-collinearity and reduce the variance of the estimator, at the cost of some bias. The question then is how to select the subset?

Definition 2.4.2. (Best subset regression) Best subset regression finds for each $k \in \{0, 1, \cdots, p\}$, the subset which has the smallest residual sum of squares, see Def. 2.3.2.

Example (2.3.1 revisited). Going back to the house price example we have a maximum of 9 explanatory variates and the intercept term. Figure 2.4 shows the residual sum of squares for all possible subsets of $X_1, \ldots, X_9$ - the intercept was always included – plotted against k, the number of terms in the subset. The red curved is the *best subset curve* and defines the models which are eligible for selection in the best subset method. Since this curve must be a decreasing one – adding any variable always gives a 'better' fit, hence

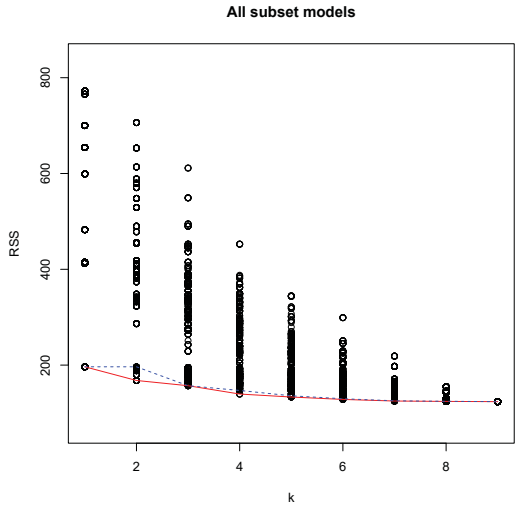

Figure 2.4: RSS for all size k subsets. Red solid lines is best subset curve, blue dashed line is greedy forward selection

reduces RSS – it can not be used for finding the 'optimal' value for k. How to select k involves the tradeoff between bias and variance, as described in §2.4.1. Typically the smallest model is selected which minimizes an estimate of expected prediction error, see §2.5.

Looking at all possible subsets and fitting a model to each choice can be computationally very expensive. In Example 2.3.1 there are $2^9 = 512$ possible models so it is *just* about possible to do in a reasonable time. In general we use non-exhaustive search methods which are computational much faster but may not find the 'best' model however commonly find ones close to the best.

Most method we look at will, like best subset regression produce a set of possible models indexed by a single parameter – in this case k. The job is then the find the 'right' value of this parameter.

2.4.3 Forward and backward step selection

Definition 2.4.3. (Forward step selection) This is a greedy algorithm which starts with the intercepts then adds the covariate which most decreases the RSS at each stage. Rather than explore the whole of the space of possible models it looks for a 'good' yet easy to compute path through them.

Example (2.3.1 revisited). In Fig. 2.4 the blue dashed line shows the path, in terms of RSS and k, of the forward selection algorithm. Here its is

close to the best subset curve for most values of k, but there is no guarantee that that is always going to happen.

Definition 2.4.4. (Backward step selection) Starts with the full model and deletes the covariate with the smallest z-statistic

Definition 2.4.5. (Akaike information criterion) The Akaike information criterion (AIC) tries to balance goodness of fit of a model and the models complexity. It measures the complexity of the model by counting the number of parameters used. It is defined as

$$-2\ell(\widehat{\theta}) + 2N_p$$

where ℓ is log-likelihood, $\widehat{\theta}$ the MLE and N_p the number of parameters in model.

In general a smaller value of AIC is preferred. As you add parameters – make the model more complex – $-2\ell(\widehat{\theta})$ gets smaller but the penalty term, $2N_p$, counts against adding parameters which don't improve the fit enough

Definition 2.4.6. (The `step( )` function) In R we have the function `step( )` whose default is to use both a forward and backward one step search at each stage – i.e. add any one variable not currently included and delete any one variable that has been included. The decision to add or drop is not done by just looking at the RSS but by using the AIC criterion which takes into account the complexity of the model.

2.4.4 Computing in R

We can do all the previous calculations in R very easily, but sometimes we need to add packages to the basic R framework. For example consider

```
> library(MASS)
> stepAIC(lm(Y ~X1+X2+X2+X3+X4+X5+X6+X7+X8+X9, data=house.price) )
```

This opens the library `MASS` and uses the function `stepAIC( )` from it. This function allows forward, backward (and both) stepwise searches through model space. If you get the response

```
> library(MASS)
Error in library(MASS) : there is no package called 'MASS'
```

when you try this you would need to use the *Package Installer* to download the library from *CRAN* before you start.

If p is not too large you can try the exhaustive search method. This can be done very efficiently with the code

```
> library(leaps)
> least.subs <- leaps(X,y, nbest=1)
```

2.4.5 Ridge regression

Subset selection methods take a very binary approach, the explanatory variable is either included or excluded. We can look at more continuous ways of adapting the model – always using a tuning parameter akin to k – of changing the explanatory information.

Definition 2.4.7. (Ridge regression) The ridge regression estimate of a linear model is defined as

$$\widehat{\beta}^{ridge} := \arg\min_{\beta} RSS(\lambda),$$

where

$$RSS(\lambda) := \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2.5}$$

where $\lambda > 0$ is a tuning parameter which determines the bias-variance trade-off.

This problem can be written in an equivalent form as

$$\begin{aligned} \widehat{\beta}^{ridge} \quad := \quad & \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ & \text{s.t.} \quad \sum_{j=1}^{p} \beta_j^2 \leq t \end{aligned} \tag{2.6}$$

where there is a one-to-one correspondence between λ and t.

Without loss for the predictive power of the model we usual centre the explanatory variables such that $\sum_{i=1}^{n} x_{ij} = 0$. When the centring has been done we estimate β_0 with $\bar{y}$. Let $\boldsymbol{X}$ be the $N \times p$ matrix whose columns are the centred covariates and we don't include the constant term. With this definition we have

$$RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda \beta^T \beta$$

and here $\beta = (\beta_1, \ldots, \beta_p)$. We can show that

$$\widehat{\beta}^{ridge} = (\boldsymbol{X}^T \boldsymbol{X} + \lambda I)^{-1} \boldsymbol{X}^T y.$$

From the above formula we see that if $\boldsymbol{X}^T \boldsymbol{X}$ is close to being singular –i.e. collinearity is a problem – the λI adjustment moves the matrix away from a singularity towards the identity matrix.

Example (2.3.1 revisited). We can see the effect of choosing different values of λ on the parameter values in the house example in Fig. 2.6. As λ gets bigger all parameters shrink towards zero, but at different rates. We see for example the coefficient β_6 of the 'Number of rooms' goes quickly to zero and this is the estimate that seemed to have the 'wrong' sign in the standard OLS estimates.

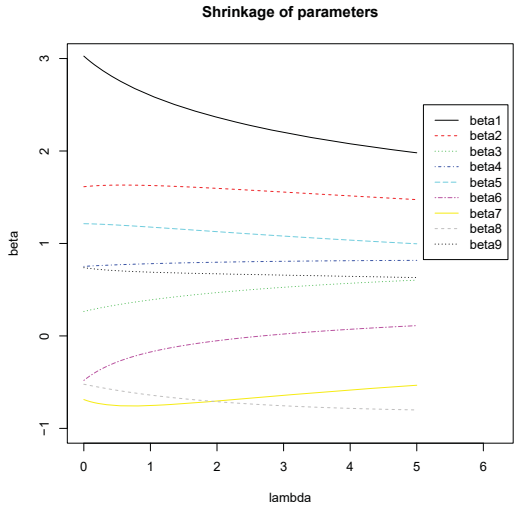

Figure 2.5: Shrinkage of parameters in house example

Definition 2.4.8. (Effective degrees of freedom) The *effective degrees of freedom* of a ridge estimate of the centred explanatory variables $\boldsymbol{X}$ is defined by

$$df(\lambda) := tr\left[\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda I)^{-1}\boldsymbol{X}^T\right].$$

When $\lambda = 0$ we get the normal degrees of freedom in the regression model i.e. p. The effective degrees of freedom measures how much shrinkage has taken place

2.4.6 The Lasso

Definition 2.4.9. (Lasso) The lasso regression estimate is defined by

$$\begin{aligned} \widehat{\beta}^{lasso} &:= \arg\min_{\beta} \sum_{i=1}^{N}(y_i - \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ &\text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \leq t \end{aligned} \tag{2.7}$$

which can be compared to Equation (2.6) where we see the constraint is on the absolute value of the parameter values not the squares. This can then also be expressed, using Lagrange multipliers as,

$$\widehat{\beta}^{lasso} := \arg\min_{\beta} \left\{ \sum_{i=1}^{N}(y_i - \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\},$$

The lasso not only shrinks small estimate to zero but also can make them exactly zero. Thus it is able to perform subset selection as well as shrinkage.

Example (2.3.1 revisited). We can see the strinkage effects of the lasso in Fig. 2.6. The left hand plot shows the lasso as λ moves from 0 (on the right of the plot) to ∞ (on the left). We see the values of the (standardised) parameter estimates as they shrink and then go to zero. For example following the red dashed line, which corresponds to β_2 is the second to last to be set to zero (i.e. when $k = 1$). The path for β_6 (solid magenta) is the first to be shrunk completely to zero.

To compare this we have in the righthand panel the forward stepwise selection, where we can see the covariates being added in the order $1, 2, 9, 8, 5, 7, 4, 3, 6$

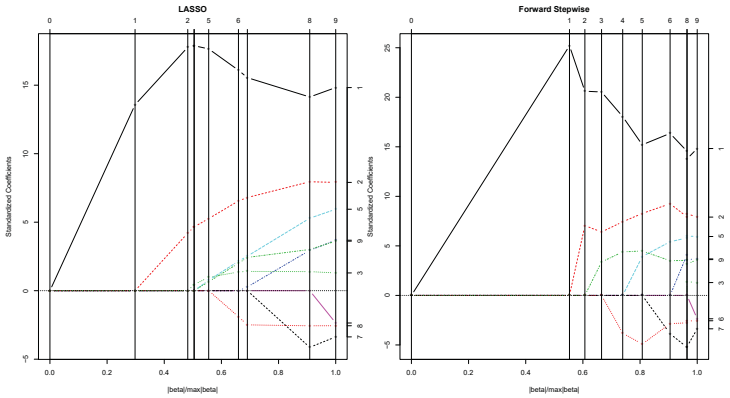

Figure 2.6: Shrinkage of parameters in house example

2.4.7 Computing in R

The following code shows how to implement ridge regression and the LASSO in R.

```
############################
#Ridge regression
> library(MASS)
> fit.ridge <-  lm.ridge(Y ~X1+X2+X2+X3+X4+X5+X6+X7+X8+X9, lambda=1, data=house.price, y

###################
#Lasso
> library(lars)
> house.lasso <- lars(X1, y, type="lasso")
```

where $X1$ is the matrix of centred covariates.

2.5 Model assessment and selection

The previous section describes how we can look at the model complexity of regression models in terms of one dimensional tuning parameters: k the number of covariates in subset selection, λ a positive real number in ridge regression or the lasso. We have not yet discussed how to select the 'best' value of such a tuning parameter for the prediction problem. We emphasis that finding the model which best fits the training data is not going to be the best model for forecasting because of the bias-variance trade-off. Very complex models can fit training data very well but have associated very high variance. Another way of viewing this issue is to recall that for prediction we want a model that fits the underlying population, but by fitting very complex models to a sample just getting a very good description of the sample. Often a complex model will not be able to generalise away from the observed sample – we say it has over fitted, describing detailed aspects of the sample that are there just by 'chance'.

Let use define how we can evaluate a models predictive performance in a way that has taking into account all aspects of the model fitting process.

Definition 2.5.1. (Generalization error) Let $\mathcal{T}$ be the training data, i.e. for regression the observed set $\{(y_1, \boldsymbol{x}_1), \ldots, (y_N, \boldsymbol{x}_N)\}$ where N is the number of cases, y_i the response and $\boldsymbol{x}_i$ the covariates for the i^{th}-case.

Let $\widehat{f}(\boldsymbol{X_0})$ be the prediction model fitted to the training data and evaluated at value of the covariate selected from the population, $\boldsymbol{X_0}$. We define the *generalization error* to be

$$Err_{\mathcal{T}} := E\left[L(Y, \widehat{f}(\boldsymbol{X_0}))|\mathcal{T}\right].$$

where $L()$ is any loss function, but in this section will typically be squared loss i.e. the generalization error is $E\left[\left(Y - \widehat{f}(\boldsymbol{X_0})\right)^2 |\mathcal{T}\right]$. In this we think of Y and X_0 as both being drawn from their joint distribution.

We can also define the *expected generalization error* to be $Err_{\mathcal{T}}$ averaged over training sets

$$Err := E_{\mathcal{T}}\left[Err_{\mathcal{T}}\right].$$

When we are evaluating how well a model predicts we want to compute – or at least estimate – either $Err_{\mathcal{T}}$ or, in practice, Err.

Definition 2.5.2. (Training error) The training error in general is defined as

$$\overline{err} := \frac{1}{N}\sum_{i=1}^{N} L(y_i, \widehat{f}(\boldsymbol{x}_i)),$$

which is the residual sum of squares (RSS) typically in this section.

In general training error is not a good estimate of prediction error and typically underestimates it.

Definition 2.5.3. (Ideal procedure) Hastie et al. (2009, p. 222) state that if we had lots of data, and a set of possible models, what we would ideally do is split the data into three different parts: training, validation and test.

1. Use the training to fit each of our set of models i.e. estimate their parameters.
2. Use the validation data to estimate the prediction error for each of our candidate fitted models in order to pick the best. Note here we are not reusing data to both fit and evaluate.
3. Finally, use the test data to estimate the prediction error for our selected single best model.

They propose a rough rule that 50% of data be training, 25% validation and 25% being testing.

This is however often not possible since data is scarce, so what should we do?

2.5.1 Bias-variance decomposition

Definition 2.5.4. (Expected prediction error regression) We have a regression model of the form $Y = f((X)) + \epsilon$, where $Var(\epsilon) := \sigma_\epsilon^2$ and the fitted value is $\widehat{f}(\boldsymbol{x}_0)$ for some input $\boldsymbol{X}_0 = \boldsymbol{x}_0$. Then the expected prediction error is

$$\begin{aligned} Err(\boldsymbol{x}_0) &= E\left[\left(Y - \widehat{f}(\boldsymbol{x}_0)\right)^2 | \boldsymbol{X}_0 = \boldsymbol{x}_0\right] \\ &= \sigma_\epsilon^2 + \left\{E\left[\widehat{f}(\boldsymbol{x}_0) - f(\boldsymbol{x}_0)\right]\right\}^2 + E\left[\left(\widehat{f}(\boldsymbol{x}_0) - E[\widehat{f}(\boldsymbol{x}_0)]\right)^2\right] \\ &= \sigma_\epsilon^2 + Bias^2 + Variance \end{aligned}$$

The first term is called the *irreducible error* associated with the 'true' model, f, and will always be there, the second and third terms depend on the model class we use.

Example 2.5.5. (Simulation Example) In order to understand the generalisation error for models selected by best subset regression a simulation experiment was done. The true model is given by

$$Y = 1.0X_1 + 0.775X_2 + 0.550X_3 + 0.325X_4 + 0.100X_5 + \epsilon \tag{2.8}$$

where $\epsilon \sim N(0, 0.3^2)$. The set of possible covariates X_i is $p = 20$ dimensional and we generate each independently from $Unif(0, 1)$ distribution. A training

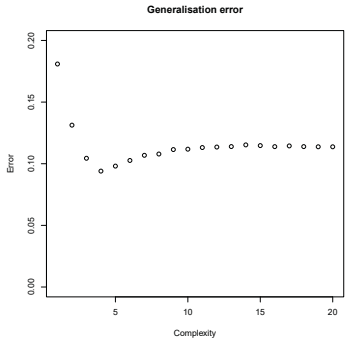

Figure 2.7: Generalisation error

sample, $\mathcal{T}$ with $N = 80$, was generated and the best subset regression was fitted for each subset size $k = 1, \ldots, 20$.

From the training set we have 20 possible models to choose from. We see how well each does in terms of its $err_{\mathcal{T}}$ prediction error by randomly selecting sets of explanatory variables (each independent $Unif(0, 1)$) generating the observed value of Y from model (2.8) and then seeing how well each of the 20 candidate fitted models did at predicting Y in terms of squared error.

With fixed $\mathcal{T}$ this was repeated many times and the average error (i.e., $err_{\mathcal{T}}$) is plotted in Fig. 2.7 We see that when the model is too simple – has less than the 5 covariances in the true model – its does very badly. It is strongly biased. But having the model too complex is not a good thing since the prediction variance starts to increase.

The simulation based approach of Example 2.5.5 does clarify what happens in model building and the trade off between bias and variance, but it can't be used in practice with a real data set since it required knowing the true model. If we try to use the training error (Definition 2.5.2) we find that, in the words of Hastie et al. (2009, p. 229) it is is too optimistic, it underestimates the true error.

Definition 2.5.6. (In-sample error) The *in-sample error* is defined by

$$Err_{in} := \frac{1}{N} \sum_{i=1}^{N} E_{Y^0} \left[L(Y^0, \widehat{f}(\boldsymbol{x}_i) | \mathcal{T} \right].$$

That is we fixed the values of the explanatory variables at their training values and then average over the random responses Y^0.

We define the *optimism* as being

$$op := Err_{in} - \overline{err},$$

and *average optimism* as being

$$\omega := E_Y(op).$$

The Akaike information criterion, Definition 2.4.5 can thought of as an estimate of Err_{in} and we can use it for model selection we simply choose the model giving smallest AIC over the set of models considered.

Example (2.3.1 revisited). We can use the `stepAIC` function in the `MASS` package in R to do a forward and backwards search which stops when the smallest AIC value is found.

```
> library(MASS)
>  stepAIC(lm(Y ~X1+X2+X3+X4+X5+X6+X7+X8+X9, data=house.price) )
         [  ...  ]
       Df Sum of Sq    RSS    AIC
<none>              139.60 52.257
- X5    1    20.836 160.43 53.596
- X7    1    21.669 161.27 53.720
- X2    1    47.409 187.01 57.274
- X1    1   156.606 296.20 68.312

Call:
lm(formula = Y ~ X1 + X2 + X5 + X7, data = house.price)

Coefficients:
(Intercept)           X1           X2           X5           X7
     13.621        2.412        8.459        2.060       -2.215

>    summary(lm(formula = Y ~ X1 + X2 + X5 + X7, data = house.price))

 Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.6212     3.6725   3.709 0.001489 **
X1            2.4123     0.5225   4.617 0.000188 ***
X2            8.4589     3.3300   2.540 0.019970 *
X5            2.0604     1.2235   1.684 0.108541
X7           -2.2154     1.2901  -1.717 0.102176
```

Example (2.5.5 revisited). If we look again at the simulated exercise we can see how well the AIC based search works.

```
 > stepAIC(lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10))
    [  ...  ]

Call:
```

```
lm(formula = y ~ x1 + x2 + x3 + x4)

Coefficients:
(Intercept)           x1           x2           x3           x4
     0.1332       0.9902       0.8383       0.4564       0.2637
```

It indeed picks out the $p = 4$ model that Fig. 2.7 shows has the best prediction error, but of course unlike that figure it has not used any knowledge of the true model to do it.

Definition 2.5.7. (AIC(α)) Given a set of candidate models which have be indexed by a tuning parameter α we can define

$$AIC(\alpha) = \overline{err}(\alpha) + 2\frac{d(\alpha)}{N}\widehat{\sigma}_\epsilon^2$$

where $\overline{err}(\alpha)$ is the training error and $d(\alpha)$ the number of parameters (or the effective degrees of freedom) for the model indexed by α. This function provides an estimate of the test error curve, and we find the tuning parameter $\widehat{\alpha}$ that minimizes it.

2.5.2 Cross validation

For most examples the 'ideal procedure' of Definition 2.5.3 requires too much data but it can be thought of as a way of motivating the most popular method for estimating prediction error, this is called cross validation. It works in a very wide range of prediction problems, but here we will look at it in the regression context.

Definition 2.5.8. (K-fold cross-validation) The idea is to split the N subjects in the training set into K (roughly) equal-sized parts, say $\{\mathcal{T}_k\}_{k=1}^K$. For $k = 1, \ldots, K$ take out the k^{th}-part of this partition, $\mathcal{T}_k$ and fit the model to the other $K - 1$ parts of the data. Calculate the prediction error of the fitted model for $\mathcal{T}_k$. The prediction error is then estimate by averaging over the K times this can be done.

Example (2.5.5 revisited). Suppose we want to evaluate the 20 models which were proposed by the best subset regression the simulation exercise Example 2.5.5. Best subset regression gives a set of models, indexed by k, but does not tell you which k to use. The cross validation method can be used here to estimate k. In Fig. 2.8 one case was left out at a time – i.e. $K = N$ cross-validation – the 20 possible models were fitted to the remaining data and the error each makes in predicting the remaining data was recorded. We average this over all N cases.

We see very similar results to Fig. 2.7 but the cross validation only uses the data which is available to the analyst. It does not use any knowledge of the 'true model'.

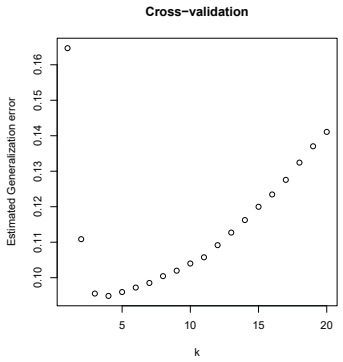

Figure 2.8: Cross validation estimate of generalisation error.

One difference, though, between Figs. 2.7 and 2.8 is the type of generalisation error being estimated. In general cross-validation gives us an estimate of the expected generalization error err and not the generalization error, $err_{\mathcal{T}}$, itself. Since, in a given prediction problem we have one specific training set, it would be more appropriate to have a good idea about $err_{\mathcal{T}}$, but in general this is not an easy thing to estimate.

2.6 Appendix: Review of Regression Theory

2.6.1 Matrix differentiation review

- Let $\mathbf{x} = (x_1, \ldots, x_p)^T$ be a p-dimensional vector and $\mathbf{y} = \mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_q(\mathbf{x}))^T$ a q-dimensional vector.
- Define the derivative of $\mathbf{x}$ as the following $p \times q$ matrix

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = \left(\frac{\partial g_i(\mathbf{x})}{\partial x_j} \right)_{i,j} .$$

- If $\mathbf{y} = \mathbf{A}\,\mathbf{x}$, for a constant matrix $\mathbf{A}$, then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A} .$$

- If $\mathbf{y} = \mathbf{x}^T\,\mathbf{A}$, then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A}^T .$$

- If $\mathbf{y} = \mathbf{x}^T\,\mathbf{A}\,\mathbf{x}$, then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{x}^T\,(\mathbf{A} + \mathbf{A}^T) .$$

2.6.2 More linear regression results

Assuming the model $Y_i = \beta_0 + \beta_1 x_{i\,1} + \cdots + \beta_p x_{i\,p} + \epsilon_i$. We may wish to test

$$H_0 \,:\, \beta_1 = \beta_2 = \cdots = \beta_p = 0$$
$$H_1 \,:\, \text{ at least one } \beta_i \neq 0,\ i = 1, \ldots, p\,.$$

Under the assumption of normality, the test statistic

$$F = \frac{\text{SSReg}/p}{\text{RSS}/(n-p-1)}$$

has an F distribution with p and $n-p-1$ degrees of freedom, i.e. $F(p,\, n-p-1)$. Reject H_0 if $F > F_{\alpha;\, p,\, n-p-1}$, where $F_{\alpha;\, p,\, n-p-1}$ is upper α quantile of $F(p,\, n-p-1)$ distribution.

Analysis of variance table

Source of variation	Degrees of freedom (df)	Sum of squares (SS)	Mean squares (MS)	F statistic
Regression	p	SSReg	SSReg/p	$F = \frac{\text{SSReg}/p}{\text{RSS}/(n-p-1)}$
Residual	$n-p-1$	RSS	$\widehat{\sigma^2} = \text{RSS}/(n-p-1)$	
Total	$n-1$	TSS		

The F-test is used to test that there exists a linear relationship between Y and p covariates in $\mathbf{x}$ vector. If it iss significant then a natural question is for which of the p covariate in $\mathbf{x}$ there is an evidence of a linear relationship with Y. To answer this question, one can perform p separate t-tests using the test statistics

$$T_i = \frac{\widehat{\beta}_i - \beta_i}{\text{se}(\widehat{\beta}_i)} \qquad i = 1, \ldots, p\,,$$

where under $H_0 \,:\, \beta_i = 0$, $T_i \sim t_{n-p-1}$, t distribution with $n-p-1$ degrees of freedom. The quantity $\text{se}(\widehat{\beta}_i)$ is the estimated standard deviation of $\widehat{\beta}_i$. Note that, for $i = 0,\, 1,\, \ldots,\, p$, $\text{se}(\widehat{\beta}_i)$ is the square root of the $(i,\, i)$ element of the matrix $\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}] = \widehat{\sigma}^2\,(\mathbf{X}^T\mathbf{X})^{-1}$. A $(1-\alpha)100\%$ confidence interval for β_i is

$$\widehat{\beta}_i - t_{(\alpha/2,\, n-p-1)}\text{se}(\widehat{\beta}_i) \leq \beta_i \leq \widehat{\beta}_i + t_{(\alpha/2,\, n-p-1)}\text{se}(\widehat{\beta}_i)\,.$$

How to test presence of a specified subset of the predictors in the model? This means, for $1 \leq s \leq p$, testing the null hypothesis

$$H_0 :\, \beta_1 = \beta_2 = \cdots = \beta_s = 0\,, \quad \text{versus } \mathrm{H}_1 : H_0 \text{ not true.}$$

That is, testing the reduced model $Y = \beta_0 + \beta_{s+1}x_{s+1} \cdots + \beta_p x_p + \epsilon$, versus the full model $Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_s x_s + \beta_{s+1}x_{s+1} \cdots + \beta_p x_p + \epsilon$. The

partial F-test statistic is used

$$F = \frac{(\text{RSS(reduced model)} - \text{RSS(full model)})/(df_{\text{reduced}} - df_{full})}{\text{RSS(full model)}/df_{\text{full}}}$$
$$= \frac{(\text{RSS}(\widehat{\beta}_0, \widehat{\beta}_{s+1}, \ldots, \widehat{\beta}_p) - \text{RSS}(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_s, \widehat{\beta}_{s+1}, \ldots, \widehat{\beta}_p))/s}{\text{RSS}(\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_s, \widehat{\beta}_{s+1}, \ldots, \widehat{\beta}_p)/(n-p-1)}$$

Reject H_0 if $F > F_{\alpha;\, s,\, n-p-1}$

To compare different fits, one may use *the coefficient of determination* of the regression fit

$$\mathrm{R}^2 = \frac{\text{SSReg}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}\,.$$

A major deficiency of R^2 is that adding a predictor to the model (even irrelevant) always increases R^2.

$$\text{RSS(fit of small model)} \geq \text{RSS(fit of larger model)}\,.$$

To compensate for this an adjusted coefficient of determination is proposed.

$$\mathrm{R}^2_{\text{adj}} = 1 - \frac{\text{RSS}/(n-p+1)}{\text{TSS}/(n-1)} = 1 - \left(\frac{n-1}{n-p-1}\right)(1-\mathrm{R}^2)\,.$$

When comparing models with different numbers of predictors one should use $\mathrm{R}^2_{\text{adj}}$ and not R^2.

3

Forecasting Stationary processes

This chapter looks at stationary time series and how to forecast them. It looks in particular at how, by estimating moment structures – such as mean vectors and variance-covariance matrices – optimal linear predictors can be computed. It shows how certain models have moment structures which are estimable with the data available to us, and discusses how to select, fit and check these models in practice.

The stationary time series studied will form the basis of a much richer class of models which will have application to 'real-world' forecasting. How this is done is shown in Chapter 4.

3.1 Introduction

Definition 3.1.1. (Time series and finite dimensional distributions) A *discrete time series* is a set of random variables, $\{X_t\}$, indexed by $t \in T \subseteq \mathbb{Z}$. For any finite subset of $\{t_1, \ldots, t_n\} \subset T$ the *finite dimensional distributions* are the joint distributions

$$F(x_{t_1}, \cdots, x_{t_n}) := P(X_{t_1} \leq x_{t_1}, \cdots, X_{t_n} \leq x_{t_n}).$$

Example 3.1.2. (An i.i.d. sequence) A sequence of independent identically distributed (i.i.d.) random variables $\{X_t\}$, such that $X_t \sim F_X$ for all t, form a discrete time series, with

$$F(x_{t_1}, \cdots, x_{t_n}) = \prod_{i=1}^{n} F_X(x_{t_i}).$$

Example 3.1.3. (White noise) A sequence of uncorrelated random variables, $\{X_t\}$, each with $E(X_t) = 0$ and $Var(X_t) = \sigma^2$, for all $t \in T$, is called a *white noise process* and is denoted by $\text{WN}(0, \sigma^2)$. For this time series

the finite dimensional distributions are not defined explicitly but the second order moment structure (i.e. all means, variances and covariances) is.

Example 3.1.4. (Gaussian process) Let $T = \mathbb{Z}$, then $\{X_t\}$ is a discrete Gaussian process if any finite subset $\{X_{t_1}, \ldots, X_{t_n}\}$ has an n-dimensional multivariate normal distribution.

This model is completely determined when the mean and variance-covariance structures are known.

Example 3.1.5. (Random walk) Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of random variables. The series defined by

$$X_t := \sum_{i=1}^{t} Z_i,$$

for $t = 1, 2, \ldots$, is called a *random walk*.

Example 3.1.6. (MA(1) process) Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of $N(0, \sigma^2)$, or more for more generality $Z_t \sim \mathrm{WN}(0, \sigma^2)$, random variables. The series defined by

$$X_t := Z_t + \theta Z_{t-1}$$

for all t, is called a *first order moving average process*, and denoted by MA(1).

Example 3.1.7. (AR(1) process) Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of $N(0, \sigma^2)$, or more for more generality $Z_t \sim \mathrm{WN}(0, \sigma^2)$, random variables. The series defined by

$$X_t = \phi X_{t-1} + Z_t$$

for all $t \in \mathbb{Z}$, is called *a first order autoregressive process* (AR(1)) *process*.

3.2 Stationary processes

Definition 3.2.1. (The auto-covariance function) If $\{X_t\}$ is a time series with $Var(X_t) < \infty$ for all $t \in T$, then the *auto-covariance function* (acvf) is defined by

$$\gamma(r, s) = Cov(X_r, X_s),$$

for $r, s \in T$.

Definition 3.2.2. (Stationarity) The time series $\{X_t\}_{t \in T}$ is said to be *stationary* if (i) $E(|X_t|^2) < \infty$ for all $t \in T$, (ii) $E(X_t) = \mu$, for all $t \in T$, and (iii) the auto-covariance function satisfies

$$\gamma(r, s) = \gamma(r + t, s + t)$$

for all $r, s, r + t, s + t \in T$.

Definition 3.2.3. (Strict Stationarity) The time series $\{X_t\}_{t\in T}$ is said to be *strictly stationary* if the finite dimensional vectors $(X_{t_1}, \cdots, X_{t_n})$ and $(X_{t_1+h}, \cdots, X_{t_n+h})$ have the same joint distributions for all finite subsets of T and all h where the translation is defined.

Example 3.2.4. (White noise process) Example 3.1.3 defines the white noise process $\mathrm{WN}(0, \sigma^2)$. This is stationary since (i) $E(|X_t|^2) = \sigma^2 + 0^2 < \infty$ for all t. (ii)$E(X_t) = 0$ for all t and (iii)

$$\begin{aligned} \gamma(r,s) &= \begin{cases} \sigma^2 & \text{if } r = s \\ 0 & \text{if } r \neq s \end{cases} \\ &= \gamma(r+t, s+t) \end{aligned}$$

for all r, s, t.

Since the finite dimensional distributions are not defined it is not strictly stationary.

Example 3.2.5. (Cauchy example) A Cauchy distributed random variable X has density function

$$f(x) = \frac{1}{\pi(1+x^2)},$$

which does not have a mean or variance. So an i.i.d. sequence $\{X_n\}$ is strictly stationary but not stationary.

Example 3.2.6. (Random walk example) The random walk, $X_t := \sum_{i=1}^{t} Z_i,$, for $T = \{1, 2, \dots\}$ was defined in Example 3.1.5. It is clear that the auto-covariance function satisfies

$$\gamma(t,t) = t\sigma^2, \tag{3.1}$$

so does not satisfy Condition (iii) in Definition 3.2.2, so $\{X_t\}$ is not stationary.

Example 3.2.7. (Predictable process) Let Z_1, Z_2 be two independent $N(0, \sigma^2)$ random variables. We can define the discrete time series

$$X_t = Z_1 \cos(2\pi t/100) + Z_2 \sin(2\pi t/100).$$

From the definition of the auto-covariance function we have

$$\begin{aligned} \gamma(r,s) &= \sigma^2 \left\{ \cos^2(2\pi r/100) + \sin^2(2\pi s/100) \right\} \\ &= \sigma^2 \cos(2\pi(r-s)/100), \end{aligned} \tag{3.2}$$

which satisfies (iii) of Definition 3.2.2. From this we can easily check that $\{X_t\}$ is stationary.

Theorem 3.2.8. (Properties of auto-covariance function) Assume $\gamma(r,s)$ is the auto-covariance function of a stationary process $\{X_t\}$, then the following statements hold.
(i) The auto-covariance function can be written as $\gamma(h) := \gamma(r+h,r)$ for all r,
(ii) $\gamma(0) \geq 0$,
(iii) $|\gamma(h)| \leq \gamma(0)$,
(iv) The auto-covariance function is an even function, i.e. $\gamma(h) = \gamma(-h)$.

Proof. (i) Since X_t is stationary $\gamma(r,s) = \gamma(r+h, s+h)$ for all h, so substituting $h = -s$ we have

$$\gamma(r,s) = \gamma(r-s,0)$$

which is just a function of one variable $h := r - s$. So, without loss we can define $\gamma(h)$ via

$$\gamma(h) := \gamma(r+h,r) = \gamma(h,0) \tag{3.3}$$

for all $h \in \mathbb{Z}$
(ii) By definition $\gamma(0) = Var(X_t) \geq 0$.
(iii) This follows from the Cauchy-Schwartz inequality,

$$|Cov(X_r, X_s)| \leq \sqrt{Var(X_r)Var(X_s)} = \gamma(0).$$

(iv) By definition

$$\gamma(-h) = Cov(X_{-h}, X_0) = Cov(X_0, X_{0+h}) = Cov(X_h, X_0) = \gamma(h).$$

□

Definition 3.2.9. (Auto-correlation function) For a stationary process $\{X_t\}$ the auto-correlation function is defined by

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}.$$

3.3 Estimable but flexible models

One class of processes where there is not a difference between strict stationarity and stationarity is the Gaussian processes first considered in Example 3.1.4. Consider the time series $\boldsymbol{X} := (X_1, \ldots, X_n)$ which a Gaussian process as a model. We have that the n-vector has a n-dimensional multivariate normal as its distribution.

Definition 3.3.1. (Multivariate normal) Let $\boldsymbol{X}$ be a n-dimensional multivariate normal random variable. Then the density of $\boldsymbol{X}$ is given by

$$(2\pi)^{-n/2} |\det \Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right\},$$

where $\boldsymbol{\mu} \in \mathrm{R}^n$ is the mean and Σ is the $n \times n$ positive-definite variance-covariance matrix of $\boldsymbol{X}$.

Now if we only assume that $\boldsymbol{X}$ has a multivariate normal model in a time series context and we want to estimate its parameters from the observed data $x_{t_1}, \ldots, x_{t_n}$ we have a problem. We have one observed vector with n-components and we want to estimate: n-parameters in μ and $\frac{1}{2}n(n+1)$ parameters in Σ. So there are far too many parameters compared to observations. We therefore are forced to restrict the model class to one which is estimable in practice.

First note that assuming stationarity helps since we have the mean vector has only one parameter,

$$\boldsymbol{\mu} = (\mu, \mu, \ldots, \mu),$$

and Σ has to be constant on off-diagonals

$$\begin{pmatrix} \sigma^2 & a & b & c & \ldots \\ a & \sigma^2 & a & b & \ldots \\ b & a & \sigma^2 & a & \ldots \\ c & b & a & \sigma^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ldots \end{pmatrix}$$

so can be specified by n parameters, and a check on positive definiteness. This gives $n+1$-parameters and n observations, still too many parameters.

We therefore are forced to make the model specification even stronger to look for parsimonious models with few parameters, but will then need to check the quality of the fit Examples include:

1. Example 3.1.2 is i.i.d. Normal which has 2 parameters μ, σ^2. This in general will not fit commonly observed real data since the independence assumption is too strong.

2. Example 3.1.6 is the MA(1) process which has, in general, three parameters μ, θ, σ^2. Its covariance structure is defined in Theorem 3.6.1.

3. Example 3.1.7 is the (AR(1) process which has, in general, three parameters μ, ϕ, σ^2, where here we need to check that the process is not a random walk, since this is not stationary. Its covariance structure is defined in Theorem 3.6.3.

3.4 Best linear predictor

We have seen from Chapter 2, §2.2 that the best predictor, in terms of mean square error, of a random variable Y given the information in a random variable X is $E(Y|X)$. To compute this requires knowledge of the joint distribution and is computationally hard for many examples. An alternative to is to look at a simpler class of predictors.

Example 3.4.1. (Best linear predictor) Consider two random variables, X and Y, with $E(X) := \mu_X, E(Y) := \mu_Y$ and all second moments are finite. We want to compute the best linear predictor of Y given X, i.e. find $\widehat{Y}(X) := a + bX$ which minimises

$$E\left(\left\{Y - \widehat{Y}(X)\right\}^2\right).$$

Minimising $L(a,b) := E\left(\{Y - (a+bX)\}^2\right)$ over a, b gives the solution. We have

$$\frac{\partial L}{\partial a}(\widehat{a},\widehat{b}) = 2E\left(\left\{Y - (\widehat{a}+\widehat{b}X)\right\}\right) = 2\left(E(Y) - \widehat{a} - \widehat{b}E(X)\right) = 0.$$

So, $\widehat{a} = E(Y) - \widehat{b}E(X)$, and substituting into $L(\hat{a},\hat{b})$ gives

$$E\left(\left\{(Y-\mu_Y) - \hat{b}(X-\mu_X)\right\}^2\right)$$

and this is minimised by $\widehat{b} = \frac{Cov(X,Y)}{Var(X)}$. Thus the best linear predictor is

$$\widehat{Y} = E(Y) + \frac{Cov(X,Y)}{Var(X)}(X - \mu_X).$$

and the MSE of the predictor is

$$Var(Y)\left(1 - Corr(X,Y)\right).$$

Note that if X and Y are uncorrelated the best linear predictor of Y is just $E(Y)$ and its MSE is $Var(Y)$.

Example 3.4.2. (Non-linear prediction) Suppose $X \sim N(0,1)$ and $Y = X^2 - 1$, then we have $E(X) = E(Y) = 0$ and the best linear predictor is $\widehat{Y} = 0$, since $Cov(X,Y) = 0$. Of course, the best predictor of Y given X is $X^2 - 1$, which has a MSE of 0. So the best linear predictor might not be very good.

Example 3.4.3. (Best linear predictor) Suppose now we wish to predict Y given X_1, X_2. We apply the same method used in Example 3.4.1, first write the best linear predictor as

$$\widehat{Y} = a_0 - a_1X_2 - a_2X_1.$$

Differentiating with respect to a_0 and substituting gives that we want to minimise

$$E\left(\{(Y-\mu_Y) - a_1(X_2-\mu_{X_2}) - a_2(X_1-\mu_{X_1})\}^2\right)$$

which is minimised by $(\widehat{a}_1, \widehat{a}_2)$ being the solution to

$$\begin{pmatrix} Var(X_2) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_1) \end{pmatrix} \begin{pmatrix} \widehat{a}_1 \\ \widehat{a}_2 \end{pmatrix} = \begin{pmatrix} Cov(Y, X_2) \\ Cov(Y, X_1) \end{pmatrix}$$

i.e., when covariance is non-singular the best predictor is

$$E(Y) + \begin{pmatrix} X_2 - \mu_{X_2} & X_1 - \mu_{X_1} \end{pmatrix} \begin{pmatrix} Var(X_2) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_1) \end{pmatrix}^{-1} \begin{pmatrix} Cov(Y, X_2) \\ Cov(Y, X_1) \end{pmatrix}.$$

The methods and results of Examples 3.4.1 and 3.4.3 can be generalised to the following theorem.

Theorem 3.4.4. (Prediction operator) If $\{X_t\}$ is a stationary time series, with mean μ, and auto-covariance function $\gamma(h)$. The best linear predictor of X_{n+h} given the set $X_n, \ldots X_1$ is

$$Pred(X_{n+h}|X_n, \ldots, X_1) := \mu + (a_1, \ldots, a_n)^T \begin{pmatrix} X_n - \mu \\ X_2 - \mu \\ \vdots \\ X_1 - \mu \end{pmatrix} \tag{3.4}$$

where $\boldsymbol{a} := (a_1, \ldots, a_n)$ satisfies the equation

$$\Gamma \boldsymbol{a} = \boldsymbol{\gamma}_{(n,h)} := (\gamma(h), \ldots, \gamma(h+n-1))^T \tag{3.5}$$

where Γ is the $n \times n$ matrix with ij^{th}-element, $\Gamma_{ij} = \gamma(|i-j|)$.

The MSE is given by

$$\gamma(0) - \boldsymbol{a}^T \boldsymbol{\gamma}_{(n,h)}.$$

Theorem 3.4.4 shows that, at least as far as linear prediction for a stationary process $\{X_t\}$ is concerned, all that is required is knowledge of the mean of the process μ and the auto-correlation function $\gamma(h)$ for a suitable range of h. Often the computation issues are concerned with inverting the $n \times n$ matrix Γ.

Example 3.4.5. Figure 3.1 shows an example of a optimal linear predictor based on an observed sample with $n = 200$. In this plot the auto-covariance function was estimated (see §3.5) and the best linear predictor for $h = 1, \ldots, 20$ was computed and plotted in blue. The error associate with the forecast is shown by the 95%-prediction interval, shown in red.

It is important to have a qualitative understanding of this example. For values of h where the auto-correlation is near zero the best linear predictor is simply the (sample) mean which is shown with the horizontal dashed line. We see that, in this example, the blue line converges to this for $h \approx 20$. The error in the forecast also converges to a limit for large h. This is just determined by the variance of the sample.

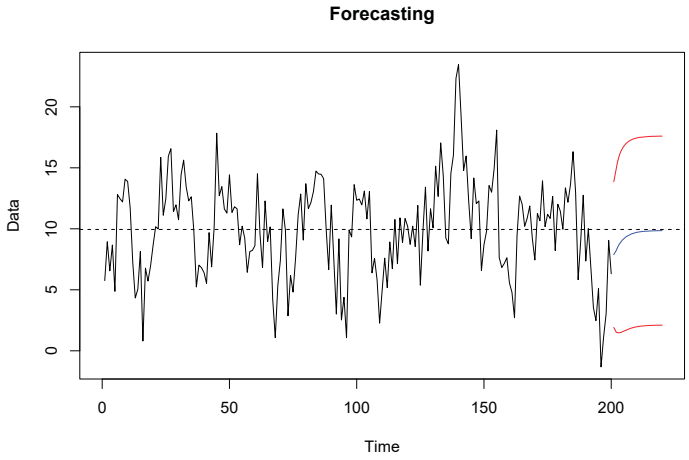

Figure 3.1: Simple forecast example: blue line forecast, red lines $\pm 1.96\times$ standard error. The dashed line is mean of process.

The time between observation and prediction before the best estimate is just the *sample mean* $\pm 1.96\times$*sample standard deviation* will depend on the shape of the auto-covariance function. If it rapidly decreases to zero then the convergence is fast. Figure 3.2 shows an example where there is a slower decay to zero. In this case we see that the convergence to the mean has not happened by $h = 20$.

3.5 Estimating the mean and auto-covariance functions

Since Theorem 3.4.4 shows that for optimal linear h-step forecasting we only need the mean and auto-covariance functions it is natural to ask if these functions can be directly estimated from observed data.

Definition 3.5.1. (Sample moments) Let $x_1, \ldots, x_n$ be observed values of a stationary time series. The *sample mean* is defined by $\bar{x} := \frac{1}{n}\sum_{t=1}^{n} x_t$, the *sample auto-covariance function* is defined by

$$\widehat{\gamma}(h) := \frac{1}{n}\sum_{t=1}^{n-|h|}\left(x_{t+|h|} - \bar{x}\right)(x_t - \bar{x}),$$

and the sample auto-correlation function is $\widehat{\rho}(h) = \frac{\widehat{\gamma}(h)}{\widehat{\gamma}(0)}$.

Since the data is not i.i.d. in general, the usual properties of the sample means do not apply, but they are still well behaved statistically.

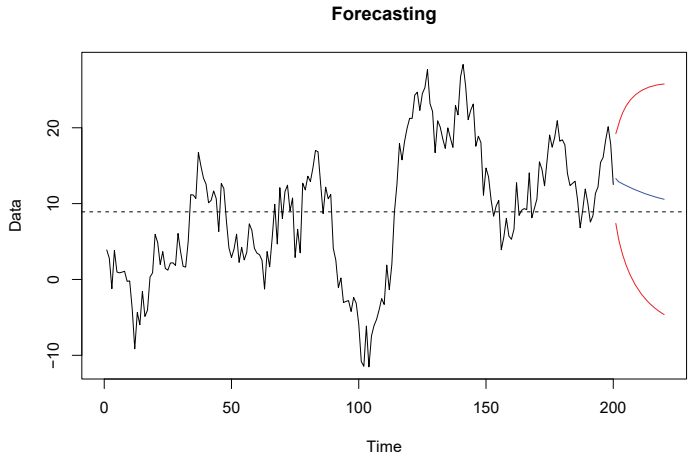

Figure 3.2: Simple forecast example: blue line forecast, red lines $\pm 1.96\times$ standard error. The dashed line is mean of process.

Theorem 3.5.2. (Properties of sample mean) For a stationary process $\{X_t\}$ the estimator $\bar{X}_n := \frac{1}{n}\sum_{t=1}^{n} X_t$ is an unbiased estimate of $E(X_t) := \mu$. Further, its mean squared error is

$$E\left(\left\{\bar{X}_n - \mu\right\}^2\right) = \frac{1}{n}\sum_{h=-n}^{n}\left(1 - \frac{|h|}{n}\right)\gamma(h).$$

If $\gamma(h) \to 0$ as $h \to \infty$, then $\bar{X}_n$ convergences, in mean square, to μ and

$$nE\left(\left\{\bar{X}_n - \mu\right\}^2\right) \to \sum_{|h|<\infty}\gamma(h),$$

if $\sum_{h=-\infty}^{\infty}|\gamma(h)| < \infty$.

Proof. We have that

$$\begin{aligned} E\left(\left\{\bar{X}_n - \mu\right\}^2\right) &= Var(\bar{X}_n) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(n - |i-j|)\gamma(i-j). \end{aligned}$$

which gives the result. □

The sample auto-covariance function, $\widehat{\gamma}(h)$, will give a biased estimate of $\gamma(h)$ – since we divide by 'n' and not '$n-p$' – but the bias is small for n much larger than h and can be disregarded. The reason for using the $1/n$ term is the following result.

Theorem 3.5.3. (Properties of sample auto-correlation function) For each $k \geq 1$ the k-dimensional sample covariance matrix

$$\widehat{\Gamma}_k := \begin{bmatrix} \widehat{\gamma}(0) & \widehat{\gamma}(1) & \cdots & \widehat{\gamma}(k-1) \\ \widehat{\gamma}(1) & \widehat{\gamma}(0) & \cdots & \widehat{\gamma}(k-2) \\ \vdots & \vdots & \cdots & \vdots \\ \widehat{\gamma}(k-1) & \widehat{\gamma}(k-2) & \cdots & \widehat{\gamma}(0) \end{bmatrix}$$

is non-negative definite.

Proof. See (Brockwell and Davis, 2002, Page 60). □

This result is important since we often want to 'plug-in' estimates to replace the unknown true covariance. We therefore want that they have the same mathematical properties such as being invertible.

Clearly if h and n are the same order of magnitude the estimate of $\gamma(h)$ will be very poor. A guideline, proposed by Box and Jenkins (1976), says that we should use $n \geq 50$ and $h \leq n/4$ if possible. When R computes the sample auto-correlation function, see Example 3.5.4 and §3.11, is has a default value for h given n which is $h = 10\log_{10}(n)$. So for $n = 100$ this gives $h = 20$, and if $n = 500$ then $h = 27$.

In using the sample auto-correlation function we need guidelines as to when an estimated correlation is 'small enough' that we could safely set it to zero.

Example 3.5.4. (Interpreting `acf()` plots) Suppose we look at i.i.d. $N(0,1)$ noise, then we know that the exact auto-correlation function is zero for all lags apart from $h = 0$. If we generate 100 observations from this model and compute the auto-correlation function we see that the estimated values are not exactly zero, for example see Fig. 3.3.

This is due to the sampling error of course. To deal with it we need to know the sampling distribution of $\widehat{\rho}(h)$ for each value of h.

It can be shown, (Brockwell and Davis, 2009, P. 222), that if the noise is i.i.d. normal then

$$\widehat{\rho}(h) \sim N\left(0, \frac{1}{n}\right).$$

In the sample acf plot in Fig. 3.3 the blue dashed lines are set at $\pm 1.96\frac{1}{\sqrt{n}}$, giving a 95%-confidence intervals for each value of h. Now, care has to be taken with interpretation when an estimate correlation lies above, or below, these lines. Even when the model is exactly i.i.d. $N(0,1)$ we would expect that 5% of the estimates would, just by chance, lie outside. An example of this is the estimate at $h = 12$.

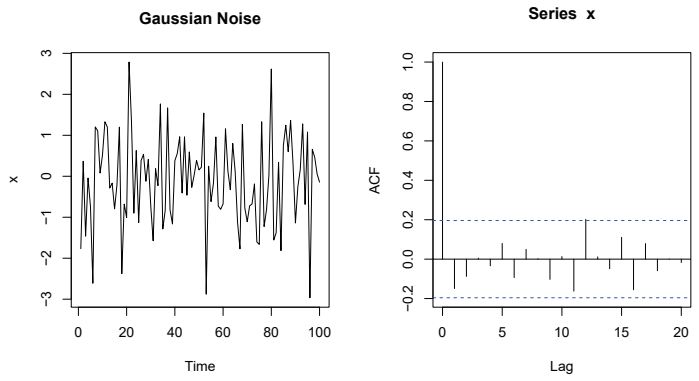

Figure 3.3: Gaussian noise and the corresponding sample auto-correlation function

3.6 Computing the auto-covariance function

Rather than estimating the mean and variance structure purely from the data we can assume a model and compute the auto-covariance function exactly. This gains efficiency when the model is correct but, of course, could be a problem when the model is incorrectly specified, see §3.10.

Theorem 3.6.1. Consider a MA(1)-process (see Definition 3.1.6) of the form $X_t := Z_t + \theta Z_{t-1}$, for $Z_t \overset{i.i.d.}{\sim} N(0, \sigma^2)$, (or $\{Z_t\} \sim \text{WN}(0, \sigma^2)$), random variables, then we have the following results:
(i) The auto-covariance function is defined by

$$\gamma(r,s) = \begin{cases} \sigma^2(1+\theta^2) & \text{for } r-s=0, \\ \sigma^2\theta & \text{for } |r-s|=1, \\ 0 & \text{for } |r-s|>1. \end{cases}$$

(ii) X_t is a stationary process for all values of θ and $\sigma > 0$.

Proof. For the first case $r = s$ we have

$$\begin{aligned} \gamma(r,s) &= Cov(X_r, X_r) \\ &= Cov(Z_r + \theta Z_{r-1}, Z_r + \theta Z_{r-1}) \\ &= Cov(Z_r, Z_r) + 2\theta Cov(Z_r, Z_{r-1}) + \theta^2 Cov(Z_{r-1}, Z_{r-1}) \\ &= \sigma^2 + 0 + \sigma^2\theta^2 \end{aligned}$$

The other cases follow the same type of argument.

(ii) Follows by directly checking the conditions of Definition 3.2.2.

□

Example 3.6.2. (MA(1) example) Figure 3.4 shows an example of the auto-correlation function for two MA(1) processes. In the left hand panel $\theta = 0.6$, while in the right panel $\theta = -0.6$. We see that after $h = 1$ the values of the auto-correlation function are exactly zero.

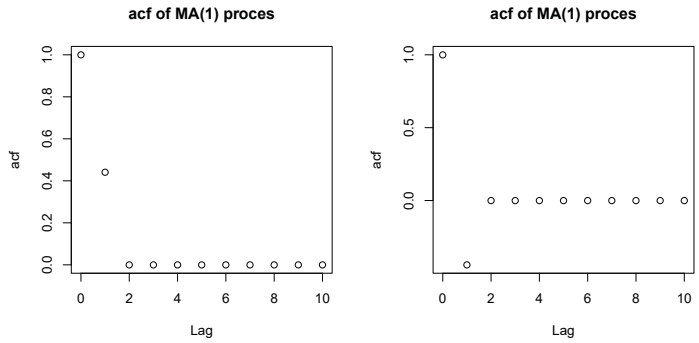

Figure 3.4: The auto-correlation function of two MA(1) processes: left panel $\theta = 0.6$, right panel $\theta = -0.6$

Theorem 3.6.3. (AR(1) example) Let $\{X_t\}$ be an AR(1) process defined by

$$X_t = \phi X_{t-1} + Z_t,$$

for $Z_t \overset{i.i.d.}{\sim} N(0, \sigma^2)$, or $\{Z_t\} \sim \text{WN}(0, \sigma^2)$, random variables.

Assuming that the process is stationary, $|\phi| < 1$ and X_t is uncorrelated with Z_{t+h} for $h > 0$, then the auto-covariance function is given by

$$\gamma(h) = \frac{\sigma^2 \phi^{|h|}}{1 - \phi^2}.$$

Proof. By definition, and for $h > 0$,

$$\begin{aligned} \gamma(h) &= Cov(X_t, X_{t-h}) \\ &= Cov(\phi X_{t-1} + Z_t, X_{t-h}) \\ &= \phi Cov(X_{t-1}, X_{t-h}) + Cov(Z_t, X_{t-h}) \\ &= \phi\gamma(h-1) + 0 \end{aligned}$$

Iterating this argument gives $\gamma(h) = \phi^h \gamma(0)$. Further,

$$\gamma(0) = Cov(X_t, X_t) = Cov(\phi X_{t-1} + Z_t, \phi X_{t-1} + Z_t) = \phi^2 \gamma(0) + \sigma^2$$

Hence, we have

$$\gamma(0) = \frac{\sigma^2}{1 - \phi^2}.$$

Combining these results gives, using the fact that $\gamma()$ is an even function, gives

$$\gamma(h) = \frac{\sigma^2 \phi^{|h|}}{1 - \phi^2}.$$

□

Example 3.6.4. (AR(1) example) Figure 3.6 shows plots of the auto-correlation function for two AR(1) models. The left hand panel has $\phi = 0.6$ and the right hand panel has $\phi = -0.6$. For Theorem 3.6.3 to hold there are regularity results that need to be checked. In fact these conditions hold in these examples as shown later in Theorem 3.8.4.

In both plots we see exponential decay, which is characteristic of the auto-correlation of AR-process. In the right hand plot the values of the correlation alternate between being positive and negative due to the sign of $\phi^{|h|}$.

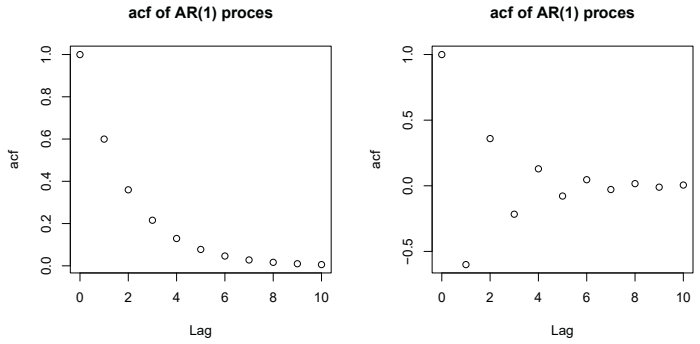

Figure 3.5: The auto-correlation function of two AR(1) process: left panel $\phi = 0.6$, right panel $\phi = -0.6$.

Example 3.6.5. (AR(1) forecast) Consider an AR(1) process with parameters $\mu = 2, \phi = 0.5, \sigma^2 = 4$ and suppose we have observed $n = 3$ observations: $x_1 = 1.48, x_2 = 3.95, x_3 = 0.38$, and we want to forecast the next observation using the results of Theorem 3.4.4.

The auto-covariance is given by

$$\gamma(h) = \frac{\sigma^2 \phi^{|h|}}{1 - \phi^2} = \frac{16}{3} \frac{1}{2^{|h|}}.$$

So that in this case

$$\boldsymbol{\gamma}_{(3,1)} = (\gamma(1), \gamma(2), \gamma(3))^T = \left(\frac{16}{3}\frac{1}{2}, \frac{16}{3}\frac{1}{2^2}, \frac{16}{3}\frac{1}{2^3} \right)^T$$

and

$$\Gamma = \begin{pmatrix} \gamma(0) & \gamma(1) & \gamma(2) \\ \gamma(1) & \gamma(0) & \gamma(1) \\ \gamma(2) & \gamma(1) & \gamma(0) \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 16 & 8 & 4 \\ 8 & 16 & 8 \\ 4 & 8 & 16 \end{pmatrix}, \Gamma^{-1} = 16^{-1}\begin{pmatrix} 4 & -2 & 0 \\ -2 & 5 & -2 \\ 0 & -2 & 4 \end{pmatrix}$$

and we get

$$\Gamma^{-1}\boldsymbol{\gamma}_{(3,1)} = (0.5, 0, 0)^T.$$

and the forecast for X_4 would be

$$\mu + (0.5, 0, 0)\begin{pmatrix} 0.38 - \mu \\ 3.95 - \mu \\ 1.48 - \mu \end{pmatrix} = 1.19.$$

Thus in this example the forecast of X_4 only depends on the value of x_3 and not on x_1 and x_2.

Example 3.6.6. (MA(1) forecast) Consider an MA(1) process with parameters $\mu = 2, \theta = 2, \sigma^2 = 2$ and suppose we have observed 10 observations:

$$x_1, \ldots, x_{10}.$$

We want to forecast the values of X_{11}. Using the methods of Theorem 3.4.4 we need the auto-covariance function which is

$$\gamma(h) = \begin{cases} \sigma^2(1+\theta^2) = 10 & \text{for } h = 0 \\ \sigma^2\theta = 4 & \text{for } |h| = 1 \\ 0 & \text{for } |h| > 1 \end{cases}$$

to forecast X_{11} use $n = 10, h = 1$

$$(\gamma(h), \ldots, \gamma(h+n-1)) = (\gamma(1), \ldots, \gamma(10)) = (4, 0, \ldots, 0),$$

and

$$\Gamma = \begin{pmatrix} 10 & 4 & 0 & 0 & \cdots & 0 \\ 4 & 10 & 4 & 0 & \cdots & 0 \\ 0 & 4 & 10 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 4 & 10 \end{pmatrix}$$

giving a forecast

$$\mu + (\gamma(1), \ldots, \gamma(10))\,\Gamma^{-1}\begin{pmatrix} x_{10} - \mu \\ x_9 - \mu \\ \vdots \\ x_1 - \mu \end{pmatrix}.$$

In this example we see that one of the problems with forecasting is that we have to invert, what might be, very large matrices. The matrix here has the structure of a, so-called, symmetric Toeplitz matrix and there are fast algorithms which can be used to compute their inverse.

3.7 MA(q) processes

Definition 3.7.1. (Backward shift operator) The *backward shift* operator B acts on a time series $\{X_t\}$ and is defined as

$$BX_t = X_{t-1}.$$

The *difference* operator ∇ is also define on $\{X_t\}$ via

$$\nabla X_t = X_t - X_{t-1} = (1 - B)X_t,$$

where 1 here represents the identity operator.

We can combine operators as 'polynomials', thus

$$B^0X_t = X_t, BX_t = X_{t-1}, B^2X_t = X_{t-2}, B^3X_t = X_{t-3}, \ldots$$

and define

$$\boldsymbol{\theta}(B) := 1 + \theta_1 B + \cdots + \theta_q B^q.$$

Thus

$$\boldsymbol{\theta}(B)X_t = X_t + \theta_1 X_{t-1} + \cdots + \theta_q X_{t-q}.$$

Definition 3.7.2. (MA(q) process) Let $\{Z_t\}$, $t \in \mathbb{Z}$ be an i.i.d. sequence of $N(0, \sigma^2)$, or $Z_t \sim \mathrm{WN}(0, \sigma^2)$, random variables. The series defined by

$$X_t := \boldsymbol{\theta}(B)Z_t = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q},$$

for all t, is called a q^{th}-order moving average, and denoted by the notation MA(q).

Theorem 3.7.3. (Moments of MA(q) process) (i) The mean of an MA(q) process is

$$E(X_t) = E\left(\boldsymbol{\theta}(B)Z_t\right) = 0$$

for all t.

(ii) The auto-covariance of an MA(q) process is given by

$$\gamma(r, s) = \begin{cases} \sigma^2 \sum_{j=0}^{q-|r-s|} \theta_j \theta_{j+|r-s|} & \text{if } |r-s| \le q, \\ 0 & \text{Otherwise.} \end{cases} \tag{3.6}$$

(iii) All MA(q) processes are stationary.

Example 3.7.4. (Auto-correlation of MA(2) process) The plots in Fig. 3.6 show the auto-correlation function $\rho(h)$ for two MA(2) processes. The left hand panel shows the case $\theta_1 = 0.6, \theta_2 = 0.4$ and the right hand panel shows the case $\theta_1 = -0.6, \theta_2 = 0.4$. Both have the property that all correlations, and hence covariances, are zero after lag 2.

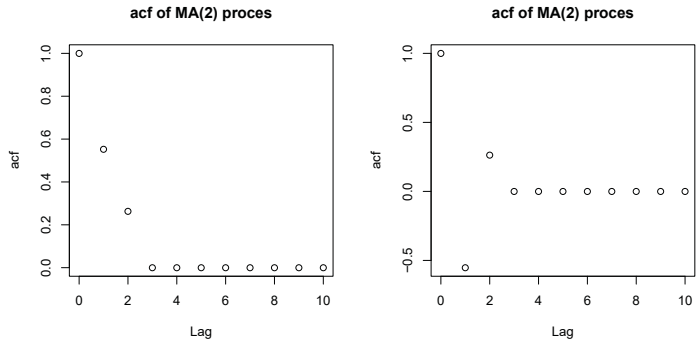

Figure 3.6: The auto-correlation function of two MA(2) processes: left panel $\theta_1 = 0.6, \theta_2 = 0.4$, right panel $\theta_1 = -0.6, \theta_2 = 0.4$.

3.8 The AR(p) process

Just as polynomials can be extended to infinite series, as long as we are careful about convergence, moving average processes can have infinite order. This is described in the Appendix 3.13.1. As an example of such an infinite moving average process consider the following argument about an AR(1) process.

Example 3.8.1. (Solving the AR(1) equation) The AR(1) process defined in Definition 3.1.7 was, in Theorem 3.6.3, assumed to be stationary when $|\phi| < 1$.

The process is defined implicitly as the solution of

$$X_t = \phi X_{t-1} + Z_t,$$

for $\{Z_t\} \sim \text{WN}(0, \sigma^2)$. We can, at least informally, write it as

$$\begin{aligned} X_t &= \phi X_{t-1} + Z_t = \phi\left(\phi X_{t-2} + Z_{t-1}\right) + Z_t \\ &= \phi^2 X_{t-2} + \left(Z_t + \phi Z_{t-1}\right) = \phi^2\left(\phi X_{t-3} + Z_{t-2}\right) + \left(Z_t + \phi Z_{t-1}\right) \\ &= \phi^3 X_{t-3} + \left(Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2}\right) \\ &\vdots \\ &= Z_t + \phi Z_{t-1} + \phi^2 Z_{t-2} + \phi^3 Z_{t-3} + \cdots \end{aligned} \tag{3.7}$$

For this to be an MA(∞) process (Definition 3.13.1) we need the coefficients to form an absolutely continuous sum. i.e. $\sum_{j=0}^{\infty} |\phi^j| < \infty$, but we have that, when $|\phi| < 1$ the standard result that

$$\sum_{j=0}^{\infty} |\phi|^j = (1 - |\phi|)^{-1} < \infty.$$

We also see that in Equation (3.7) that X_t is only a function of Z_s random variables where $s \leq t$. Hence, from the properties of $\text{WN}(0, \sigma^2)$ we have that X_t is uncorrelated with Z_{t+h} for $h > 0$, the last regularity condition of Theorem 3.6.3.

Definition 3.8.2. (The AR(p) process) Define the polynomial operator

$$\boldsymbol{\phi}(B) := 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p,$$

where B is the backward shift operator.

The $AR(p)$ process is the process which is the stationary solution to the difference equations

$$\boldsymbol{\phi}(B) X_t = Z_t \tag{3.8}$$

for $\{Z_t\} \sim \text{WN}(0, \sigma^2)$, when such a solution exists.

Definition 3.8.3. (Causal process) A causal process $\{X_t\}$ generated by $\{Z_t\} \sim WN(0, \sigma^2)$ is one where each X_t is only a function of those Z_s where $s \leq t$.

Theorem 3.8.4. (Existence of stationary solution) There exists a stationary solution to Equation (3.8) when, for $z \in \mathbb{C}$, the polynomial

$$\boldsymbol{\phi}(z) := 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p$$

has no roots which lie on the unit circle $\{z \in \mathbb{C} || z| = 1\}$.

If all roots lie strictly outside the unit circle we say there is a causal solution which can be written as

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}, \tag{3.9}$$

i.e. X_t is a function of 'previous' Z_t values.

Proof. See Brockwell and Davis (2002) or Brockwell and Davis (2009). □

Example 3.8.5. (Examples of AR(p) processes) Assume that $\{Z_t\} \sim \text{WN}(0, \sigma^2)$.

(i) Let $p = 2$, and consider

$$X_t - \frac{1}{2} X_{t-1} + \frac{1}{4} X_{t-2} = Z_t.$$

Look at the roots of the equation $1 - \frac{1}{2} z + \frac{1}{4} z^2 = 0$, since the roots, $1 \pm \sqrt{3} i$ lie outside the units circle there is a stationary solution and it has the causal form (3.9).

(ii) Let $p = 2$, and consider

$$X_t - X_{t-1} + \frac{1}{4} X_{t-2} = Z_t.$$

The corresponding polynomial, $1 - z + \frac{1}{4}z^2$, has roots, $2, 2$ which lie inside the unit circle, so there is a stationary causal solution.

(iii) Let $p = 3$, and consider

$$X_t - 5X_{t-1} + 7X_{t-2} - 3X_{t-3} = Z_t$$

The corresponding polynomial has roots $1, 1, \frac{1}{3}$, so there is not a stationary solution to these equations

Theorem 3.8.6. (Auto-covariance function for AR(p) process) If X_t is a causal, stationary AR(p) process it can be written as

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j},$$

and hence its auto-covariance function can be written as

$$\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+|h|}.$$

Example 3.8.7. (Example of AR(p) process) Figure 3.7 shows the acf for the two stationary processes. The left hand panel is an $AR(2)$ process defined by

$$X_t = 0.5X_{t-1} + 0.25X_{t-2} + Z_t,$$

and

$$X_t = 0.5X_{t-1} + 0.25X_{t-2} - 0.1X_{t-2} + Z_t,$$

where $Z_t \sim \text{WN}(0, \sigma^2)$. Numerical checks of the polynomials $\boldsymbol{\phi}(z)$ shows all roots lie outside the unit circle, so there exists stationary solutions. We see both panels show the characteristic exponential decay.

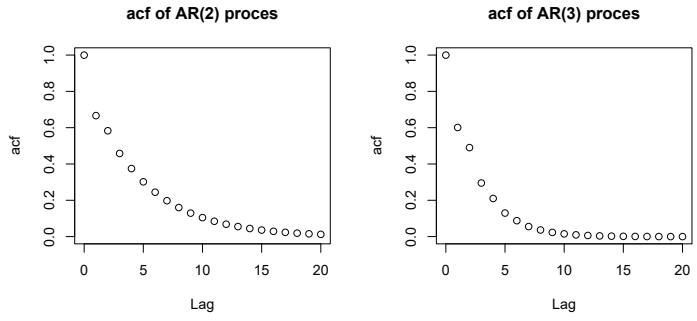

Figure 3.7: The auto-correlation function of stationary AR(p) processes: left panel $\phi_1 = \frac{1}{2}, \phi_2 = \frac{1}{4}$, right panel $\phi_1 = \frac{1}{2}, \phi_2 = \frac{1}{4}, \phi_3 = -0.1$.

3.9 The ARMA(p,q) process

Definition 3.9.1. (The ARMA process) Let $Z_t \sim \text{WN}(0,\sigma^2)$, or i.i.d. $N(0,\sigma^2)$ random variables, and define the polynomial operators

$$\boldsymbol{\phi}(B) := 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p,$$

and

$$\boldsymbol{\theta}(B) := B_t + \theta_1 B_{t-1} + \cdots + \theta_q B_{t-q},$$

where B is the backward shift operator.

The $ARMA(p,q)$ process is the process which is the stationary solution to the difference equations

$$\boldsymbol{\phi}(B)X_t = \boldsymbol{\theta}(B)Z_t \tag{3.10}$$

when such a solution exists, and we assume that the polynomials $\phi(z)$ and $\theta(z)$ share no common factors.

Theorem 3.9.2. (Existence of ARMA process) The ARMA(p,q) process defined by Equation (3.10) has a stationary solution if none of the roots of the polynomial $\boldsymbol{\phi}(z)$ lie on the unit circle $|z| = 1$. Further, if the roots lie outside the unit circle ($i.e. |z| > 1$) then there is a causal stationary solution.

3.10 Estimation of ARMA models

In this section we shall look at the problem of the estimation of an ARMA(p,q) model when p and q are considered known, and we are estimating the following parameters for, an assumed stationary, time series. The second problem, identifying p,q is discussed in §3.10.2. The parameters to be estimated are the mean μ, the p-parameters in $\boldsymbol{\phi}(B)$, the q-parameters in $\boldsymbol{\theta}(B)$ and the variance of the innovation process $Z_t \sim WN(0,\sigma^2)$, or if stronger parametric conditions are assumed, $Z_t \overset{i.i.d.}{\sim} N(0,\sigma^2)$. There are a number of ways of estimating the parameters: method of moments (Yule-Walker), least squares and, if Normal errors are assumed, likelihood methods. We concentrate on the last of these in this course.

Throughout, we will assume that we have a zero mean stationary process. This is really without loss since if $E(X_t) = \mu$, we estimate μ (Theorem 3.5.2) with the sample mean and work with the process $X_t - \widehat{\mu}$.

3.10.1 Likelihood estimation

In order to use likelihood methods for inference we need to make assumptions about the distributions involved in the ARMA(p,q) models. For simplicity we assume that the innovations $Z_t \overset{i.i.d.}{\sim} N(0,\sigma^2)$. Let us further assume that

we have observed $X_1 = x_1, \ldots, X_n = x_n$. For any ARMA$(p, q)$ process the observed $(X_1, \ldots, X_n)$ are a linear function of the unobserved

$$(Z_1, \ldots, Z_n) \sim N(0_n, \sigma^2 I_{n\times n}),$$

thus the distribution of $(X_1, \ldots, X_n)$ is also normal, and the computational issue is to compute the mean and the variance-covariance function.

Definition 3.10.1. (The likelihood function) The likelihood function for a mean zero, stationary ARMA(p, q) process is

$$Lik(\phi, \theta, \sigma^2) = \frac{1}{(2\pi)^{n/2} \det(\Gamma_n)^{1/2}} \exp\left(-\frac{1}{2} X^T \Gamma_n^{-1} X\right)$$

where Γ_n is variance covariance matrix for X of the form

$$\begin{pmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(n-2) \\ \vdots & \vdots & \vdots & \vdots \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(0) \end{pmatrix}.$$

The optimisation of the log-likelihood can not be done exactly, rather numerical methods such as the Newton-Raphson algorithm have to be used. In particular there are two non-linear functions which are involved. The first is the function which takes the parameters to the auto-covariance function,

$$\phi, \theta, \sigma^2 \to \gamma(h) \to \Gamma_n. \tag{3.11}$$

The second is the function which inverts the (potentially large) matrix

$$\Gamma \to \Gamma_n^{-1}. \tag{3.12}$$

Finally we estimate σ^2 with the residual sum of squares, as in regression.

Theorem 3.10.2. (Properties of MLE) Let the parameters of an ARMA(p, q) model be given by

$$\beta = (\phi_1, \cdots, \phi_p, \theta_1, \cdots, \theta_q),$$

and let $\widehat{\beta}$ be the maximum likelihood estimates, then for large n we have that the sampling distribution of $\widehat{\beta}$ is

$$N(\beta, n^{-1} V(\beta))$$

where $V(\beta)$ is a $(p+q) \times (p+q)$ positive definite matrix.

Rather than give the general form of $V(\beta)$ let us look at some special cases.

Theorem 3.10.3. (MLE for AR(p)) For an AR(p) model we have $V(\boldsymbol{\phi}) = \sigma^2\Gamma_p^{-1}$, so that

1. When $p = 1$
$$V(\boldsymbol{\phi}) = \left(1 - \phi_1^2\right).$$
2. When $p = 2$
$$V(\boldsymbol{\phi}) = \begin{pmatrix} 1-\phi_2^2 & -\phi_1(1+\phi_2) \\ -\phi_1(1+\phi_2) & 1-\phi_2^2 \end{pmatrix}.$$

Theorem 3.10.4. (MLE for MA(q)) For an MA(q) model two important special cases are

1. When $q = 1$
$$V(\boldsymbol{\theta}) = \left(1 - \theta_1^2\right).$$
2. When $q = 2$
$$V(\boldsymbol{\theta}) = \begin{pmatrix} 1-\theta_2^2 & -\theta_1(1+\theta_2) \\ -\theta_1(1+\theta_2) & 1-\theta_2^2 \end{pmatrix}.$$

Theorem 3.10.5. (MLE for ARMA(p, q)) For a causal and invertible ARMA(p, q) model we have

$$V(\phi, \theta) = \frac{1+\phi\theta}{(\phi+\theta)^2} \begin{pmatrix} (1-\phi^2)(1+\phi\theta) & -1(1-\theta^2)(1-\phi^2) \\ -1(1-\theta^2)(1-\phi^2) & (1-\phi^2)(1+\phi\theta) \end{pmatrix}.$$

Of course to use the likelihood approach we have to assume Normally distributed errors, if weaker assumptions are made inference can still be done: a method of moments method is called the Yule-Walker algorithm, least squares methods also exists. In general all algorithms require numerically inverting large matrices. When computing power was less than it is today algorithms such the innovations and the Durbin-Levinson algorithm have being developed, for more details see Brockwell and Davis (2002) or Brockwell and Davis (2009).

3.10.2 Identification of ARMA models

The previous section worked with the assumption that p and q in the ARMA model were known. In practice we need to go through a model selection and checking procedure just as in regression modelling. As we have seen in Chapter 2 one major problem which can seriously effect the ability of a model to forecast correctly is over-fitting. Just as in Chapter 2 penalised likelihood methods such a AIC, AICC can be very helpful in finding the

model structure. However, in general all such 'black-box' model section procedures must be used with care and the analyst should alway check that the fitted model makes sense in the context of the given problem.

Definition 3.10.6. (AIC) The Akaike information criterion (AIC) is defined as

$$AIC := -2\ell(\hat{\beta}) + 2k$$

where $\ell(\hat{\beta})$ is the maximum value of the log-likelihood for a given model, and k is the number of parameters in the model.

The AICc corrects for small samples and is given by

$$AICc := AIC + \frac{2k(k+1)}{n-k-1}.$$

For both criteria in a given set of models the preferred model is the one with the *minimum* criterion value. The idea is to penalise more complicated model (i.e. ones with more parameters) and reward simple models, as long as they fit of course.

We can also use graphical methods to get some idea of the structure of certain types of ARMA(p, q) models.

Example 3.10.7. (Using the ACF) From Theorem 3.7.3 and example in Fig. 3.6 we see that if the process is MA(q) the its auto-covariance function will be zero after lag q. Hence inspection of the sample auto-covariance (or auto-correlation) plot gives information about q.

This method will not work for an AR(p) model, for example Fig. 3.7 shows two models with different p-values having similar looking auto-correlation functions. There is a plot that will help to find p and this is called the partial auto-correlation plot, see Appendix 3.13.2 for a formal definition. Here we only seek to interpret the plot.

Example 3.10.8. (Using the PACF) Figure 3.8 shows the pacf function for the same models as in Example 3.8.7. The plots of the acf above both had the characteristic exponential decay of AR(p) processes. In contrast the pacfs, shown in Fig. 3.8 look different. The one for the AR(2)-process has non-zero only values for $h = 1, 2$ while the one for the AR(1) process has non-zero values for $h = 1, 2, 3$. This shows how the pacf can be used to give information about p in a AR(p) process.

Theorem 3.10.9. (Partial auto-correlation function) If $\alpha(h)$ is the partial auto-correlation function for a stationary AR(p) process then $\alpha(k) = 0$ for $|k| > p$.

Outside the cases of pure MA(q) and AR(p) these graphical methods can not give definite answers for model selection, indeed even in the cases that they are designed for it takes a good deal of experience (and luck) to be able to confidently find the 'correct' model.

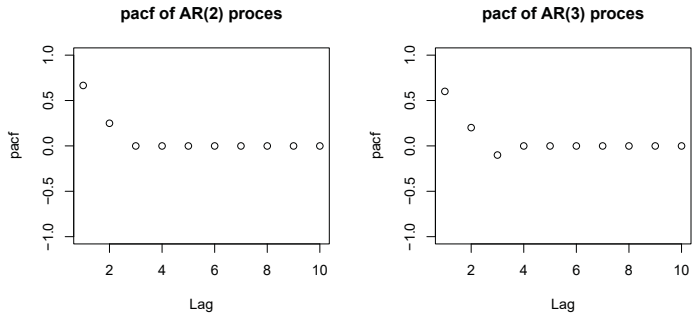

Figure 3.8: The partial auto-correlation function of stationary AR(p) processes: left panel $\phi_1 = \frac{1}{2}, \phi_2 = \frac{1}{4}$, right panel $\phi_1 = \frac{1}{2}, \phi_2 = \frac{1}{4}, \phi_3 = -0.1$.

3.11 Using R for ARMA modelling

In this section we look at some basic R, (R Core Team, 2013), commands for working with ARMA-processes. We start by simulation and plotting of realisations of ARMA models.

Example 3.11.1. (Using R) The R command:
`sim.ar2 <- arima.sim(n = 100, model=list(ar = c(0.5, 0.25)), sd = sqrt(2))`
simulates a realisation of an AR(2)-process of the form

$$\boldsymbol{\phi}(B)X_t = Z_t$$

where $Z_t \sim N(0, 2)$ and

$$\boldsymbol{\phi}(B) := 1 - 0.5B - 0.25B^2,$$

that is

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t = 0.5X_{t-1} + 0.25X_{t-2} + Z_t$$

The object `sim.ar2` is a `ts` object in R and has some built in attributes, for example

```
> print.ts(sim.ar2)
Time Series:
Start = 1
End = 100
Frequency = 1
1.48885878  4.21586053  4.06171606  3.63421248
  [ ... ]
[97]  1.05016201 -1.54411515 -0.21416876 -0.79918265
```

We can plot the time series, its auto-correlation function and its partial auto-correlation function using

```
plot.ts(sim.ar2)
acf(sim.ar2)
pacf(sim.ar2)
```

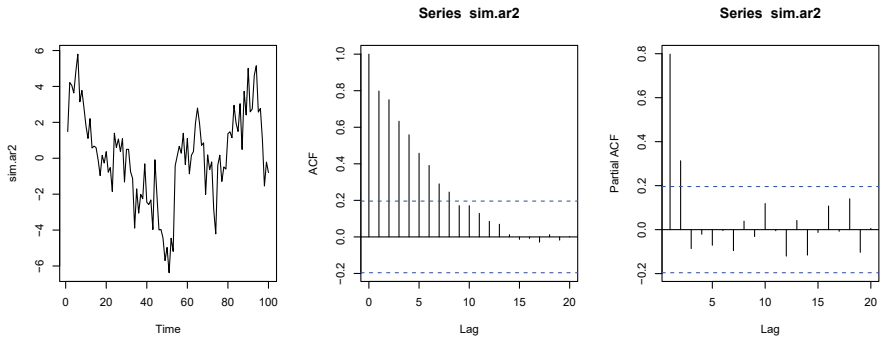

Figure 3.9: Simulated AR(2) model: the time series, its acf and pacf plots

If you try and simulate from parameter values which do not have a stationary solution you get an error message and no output.

```
> arima.sim(n = 100, model=list(ar = c(0.5, 0.5)), sd = sqrt(2))
Error in arima.sim(n = 100, model = list(ar = c(0.5, 0.5)), sd = sqrt(2)) :
  'ar' part of model is not stationary
```

Example 3.11.2. Generating a MA(q) time series is similar, for example we can generate a sample of size 500 from an MA(2) model such that

$$X_t = Z_t + 2Z_{t-1} + 5Z_{t-2}$$

where $Z_t \sim N(0, 10)$ via:

```
sim.ma2 <- arima.sim(n = 500, model=list(ma = c(2, 5)), sd = sqrt(10))
```

Example 3.11.3. For a ARMA$(1, 2)$ example we can generate a sample of size 500 from a model such that

$$X_t + 0.6X_{t-1} = Z_t + 0.6Z_{t-1} - 0.3Z_{t-2}$$

we would use

```
sim.arma12 <- arima.sim(n = 500, list(ar=c(-0.6), ma = c(0.6, -0.3)),
              sd = sqrt(1))
```

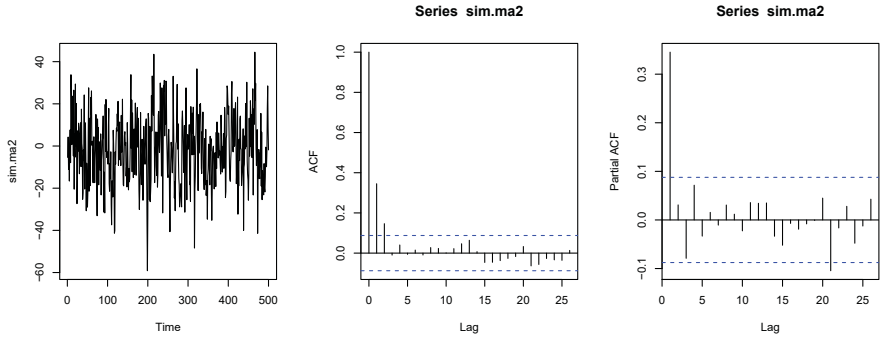

Figure 3.10: Simulated MA(2) model: the time series, its acf and pacf plots

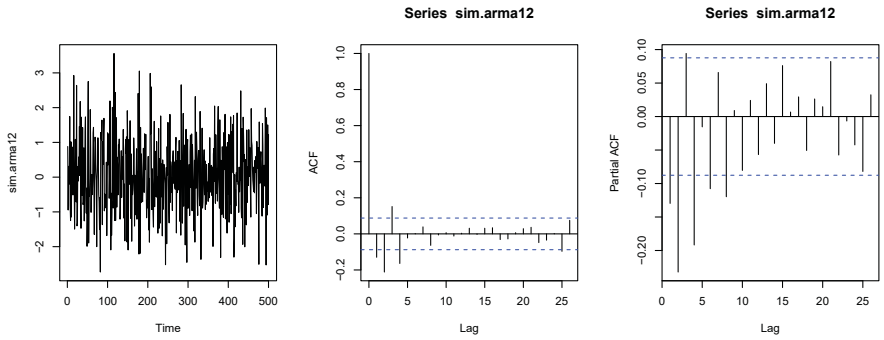

Figure 3.11: Simulated ARMA$(1, 2)$ model: the time series, its acf and pacf plots

Fitting models

If we assume, for the moment, that we know the values of p and q in the ARMA(p, q) structure then we can estimate the values of the parameters, using the `arima( )` function.

Example 3.11.4. Lets go back the the ARMA$(1, 2)$ in Example 3.11.3. We use the function

```
> arima(sim.arma12, order=c(1, 0,2), include.mean=F)

Coefficients:
          ar1     ma1      ma2
      -0.5891  0.5152  -0.3640
s.e.   0.0653  0.0701   0.0512

sigma^2 estimated as 1.015:  log likelihood = -713.57,  aic = 1435.14
>
```

From this output we see the following. First we can compare the true parameters values, which we know since this is simulated data to the estimated values and standard errors – and so 95%-confidence intervals. We see that the true values lie inside the confidence intervals and the width of the confidence intervals are quite small since this example uses a lot of data.

True	Estimated	S.E.	Confidence Interval
$\phi_1 = -0.6$	-0.5891	0.0653	(-0.72, -0.46)
$\theta_1 = 0.6$	0.5152	0.0701	(0.38, 0.65)
$\theta_2 = -0.3$	-0.3640	0.0512	(-0.46, -0.26)

In this fit we used the option `include.mean=F` since I was assuming I knew that it was a mean zero time series. If mean also had to be estimated then we can drop this option from the function call.

We can now look at the quality of the fit. Just as in regression analysis we use residuals, here estimates of the innovation process Z_t which should be white noise, or if we assume a Gaussian process, i.i.d. Normal data. We do this by the command

```
arima(sim.arma12, order=c(1, 0,2))$residual
```

which output the 500 residuals values. Again, as in regression, we can use plots to evaluate the model fit. For example in Fig. 3.12 we plot the estimated residuals, the acf plot of the residuals – which looks like white noise in this case, and a QQplot which shows the residuals do indeed look like they come from a normal distribution.

Estimating the structure

The previous section assumed that p and q are known. Of course in practice these are not known and a model selection procedure has to be undertaken.

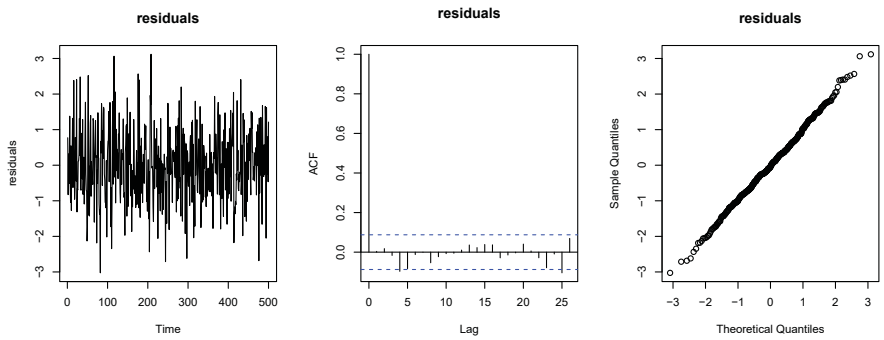

Figure 3.12: Residual plots: the time series, its acf and QQ norm

This is typically done using a mixture of graphical and numerical tools. The following rules of thumb are helpful, but should not be over interpreted. We first recall the definition of the

$$\pm 1.96\sqrt{\frac{1}{n}}$$

dashed lines which we get in R's `acf( )` and `pacf( )` output. These where discussed in Example 3.5.4 and can be thought of as pointwise 95%-confidence intervals for the hypothesis that the correlation at h is zero.

Example 3.11.5. For an MA(q) we would expect that the auto-correlation function is zero for all lags greater than q, and it can be shown that the partial auto-correlation function has exponential decay.

For example consider Fig. 3.10. The model which generated the data was MA(2) and we see that in the sample acf plots we have estimated values outside the dashed lines for $h = 0, 1, 2$. The partial auto-correlation function is trivial for all $h > 1$ apart from $h = 22$ and we can plausibly argue that is is just because we would expect 5% of the estimated values to lie outside the lies even if the true values were actually zero.

Example 3.11.6. For an AR(p) model we expect that the auto-correlation function shows exponential decay, while the partial auto-correlation function is zero for all $h > p$. We can see this in Fig. 3.8, where we have in the pacf non-zero values for $h = 0, 1$ and exponential decay in the acf plot.

For general ARMA(p, q) models unfortunately there are no simple patterns that always appear, and even the patterns described in Examples 3.11.5 and 3.11.6 tend to require quite large sample sizes to be used reliably.

The second way of doing model selection is to use the AIC or AICc values from the fit of the models. Since these use the number of the parameters the will give the same penalty to, for example an AR(2), ARMA(1, 1) and

MA(2) model. In general there will rarely be the case where the model can be uniquely identified and it is advisable to compare forecasts from different plausible models to see if there are any major differences.

Forecasting

Once a model has been fitted using the `ARIMA( )` function then the `predict( , n.ahead= )` can be used to make forecasts for h steps a head and compute the standard error of the forecast. This has already been shown in Fig. 3.1 and 3.2. These were generated with the following code: The data was a `ts`-object called `sim.forecast`, and I assumed this this came from an $AR(2)$ model, this was then fitted and its (estimated) mean and auto-covariance function was used, as in §3.4, to make a point forecast and estimate the variance around this. Forecasts for $h = 1, 2, 3$ are given by the code

```
> predict(arima(sim.ma2.forecast, order=c(2, 0,0)), n.ahead=3)
$pred
Time Series:
Start = 201
End = 203
Frequency = 1
[1] 13.30448 12.90422 12.75071

$se
Time Series:
Start = 201
End = 203
Frequency = 1
[1] 3.017765 3.780846 4.418923
```

3.12 Other stationary processes

The forecasting techniques in this chapter are designed for stationary processes of the form ARMA(p, q) where, in practice to ensure that the number of parameters is not too large

$$p + q << n.$$

These give rise to linear predictions which are basically short term – the long term all forecasts are just the mean. The length of time which counts as short-term is determined by the structure of the auto-covariance function.

We finish this chapter with a couple of examples to show that there are time series outside this parsimonious ARMA(p, q) structure which can behave quantitatively quite differently.

3.12.1 Long memory processes

A long memory process is one where the auto-covariance function decays more slowly that the exponential decay that we see in stationary ARMA-process. A process like a random walk has an auto-covariance function which does not decay at all – it is a constant. These long term memory processes therefore lie somewhere between a stationary ARMA(p, q)-process and a non-stationary random walk. They are stationary but the auto-covariance decays like a polynomial in h – that is much slower than exponentially.

Figure 3.13 shows an example of such a process. In the left hand panel a realisation of $n = 1000$ points is shown and the corresponding (sample) auto-correlation function is plotted in the right hand panel. We see the very slow decay of the function clearly.

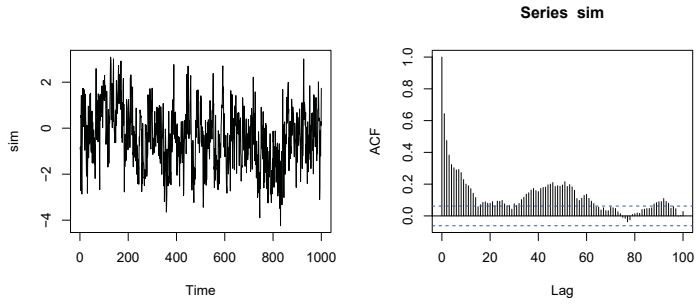

Figure 3.13: A timer series with long term dependency

3.12.2 Gaussian processes

Another class of processes which can be used in modelling are the Gaussian processes defined in Example 3.1.4. These models make explicit assumptions about the joint distribution of the observed variables, i.e. $\{X_{t_1}, \ldots, X_{t_n}\}$. Since the model is assumed Gaussian we need to defined an n-dimensional mean and an $n \times n$-dimensional variance-covariance function, and then add conditions which ensure stationarity.

One class of these is the Matern class, which has a signal parameter ν. In Fig. 3.14 four realisations of Matern class processes are shown, with $\nu = 0.5, 0.75, 1$ and 2. One thing is immediately clear: the larger the value of ν the 'smoother' the realisation.

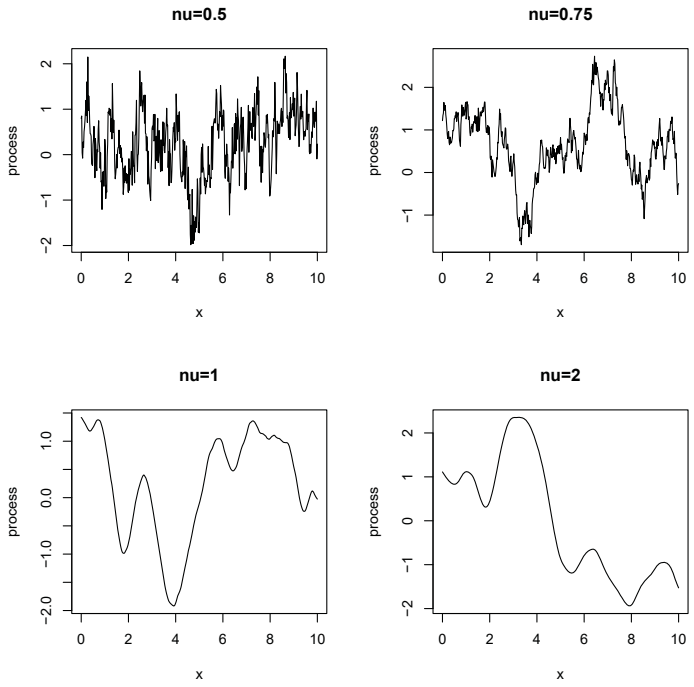

Figure 3.14: Simulations of four Matern class Gaussian process: $\nu = 0.5, 0.75, 1$ and 2. These are progressively smoother

3.13 Appendix

3.13.1 MA(∞) processes

Definition 3.13.1. Let Z_t, $t \in \mathbb{Z}$ be an i.i.d. sequence of $N(0, \sigma^2)$, (or $Z_t \sim \text{WN}(0, \sigma^2)$), random variables. Let $\{\psi_j\}$, $j = 0, 1, \cdots$ be a sequence which is absolutely convergent, i.e.

$$\sum_{j=0}^{\infty} |\psi_j| < \infty,$$

then the process defined by

$$X_t := \sum_{j=0}^{\infty} \psi_j Z_{t-j},$$

for all t, is called a infinite-order moving average, and denoted by (MA(∞)) process.

Theorem 3.13.2. The MA(∞) process of Definition 3.13.1 is stationary with zero mean and auto-covariance function

$$\gamma(h) = \sigma^2 \sum_{j=-\infty}^{\infty} \psi_j \psi_{j+|h|}.$$

Theorem 3.13.3. Wold decomposition theorem Any stationary process can be written as the sum of an MA(∞) process and an independent deterministic process, where a deterministic process is any process whose complete realisation is a deterministic function of a finite number of its values.

Example 3.13.4. The process

$$X_t = Z_1 \cos(2\pi t/100) + Z_2 \sin(2\pi t/100),$$

defined in Example 3.2.7 is a deterministic process, where Z_1, Z_2 are two independent $N(0, \sigma^2)$ random variables.

3.13.2 Partial Correlation

Definition 3.13.5. If X, Y and Z are random variables the (sample) *partial correlation* of X and Y given Z, denoted by $\rho_{xy.z}$ is the (sample) correlation between r_X and r_Y, where these are the residuals from the linear regressions of X on Z and Y on Z, respectively.

Example 3.13.6. (Adapted from Mardia et al. (1979)[page 170]) Suppose we had 20 observations on verbal skills (x), weight (y) and age (z) for a group of children. If we plot the verbal skills score against weight (see Fig. 3.15 (a)) we see a high positive correlation (0.89).

This we think is spurious and just due to the effect of age. We can control for the effect of age; regress x and y on age z. If we look at the plot of the corresponding residuals (see Fig. 3.15 (b)) we see almost no correlation left (-0.21). Panels (c) and (d) show the very strong common relationship between the two variables and age, which is giving the confounding.

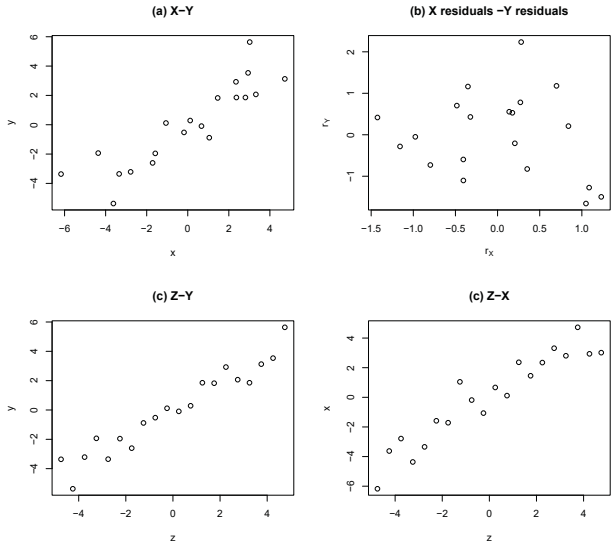

Figure 3.15: Partial Correlation: (a) scatterplot X, Y, (b) residuals, (c) Y, Z (d) X, Z

Definition 3.13.7. The partial auto-correlation function (pacf) for a stationary process, $\{X_t\}$, is defined by

$$\alpha(h) = \begin{cases} Cor(X_1, X_0) & \text{for } |h| = 1 \\ \rho_{X_h X_0 \cdot \{X_{h-1} \ldots X_1\}} & \text{for } |h| > 1 \end{cases}$$

where $\rho_{X_h X_0 \cdot \{X_{h-1} \ldots X_1\}}$ is the partial correlation of X_h and X_0 given the set $\{X_{h-1} \ldots X_1\}$.

4

The Box-Jenkins approach to forecasting and control

4.1 Introduction

The key result of Chapter 3 is Theorem 3.4.4. This states that for a stationary time series $\{X_t\}$, with mean μ, and auto-covariance function $\gamma(h)$, the best linear predictor, in terms of mean square error, of X_{n+h} based on $X_1, \ldots, X_n$ is given by

$$Pred(X_{n+h}|X_n, \ldots, X_1) := \mu + (a_1, \ldots, a_n)^T \begin{pmatrix} X_n - \mu \\ X_2 - \mu \\ \vdots \\ X_1 - \mu \end{pmatrix},$$

where $\boldsymbol{a} := (a_1, \ldots, a_n)$ satisfies the equation

$$\Gamma \boldsymbol{a} = \boldsymbol{\gamma}_{(n,h)} := (\gamma(h), \ldots, \gamma(h+n-1))^T, \tag{4.1}$$

and Γ is the $n \times n$ matrix with ij^{th}-element, $\Gamma_{ij} = \gamma(|i-j|)$.

While this is important as a theoretical result it does not have many direct applications because most observed time series are not stationary. Two key observations, though, do mean we can extend the theorem to cases of practical importances.

1. Fortunately, a large number of observed time series do fall into the class of being *linear transformations* of stationary processes. These include examples with non-constant means, strong seasonal effects, and non-constant variances, which are not themselves stationary. An example is 4.2.1.

2. Theorem 3.4.4 only uses linear functions, means and covariances and we know how all these behave under linear transformations. This allows a powerful generalisation, Theorem 4.1.1.

These generalisations do not come completely for 'free', and some extra assumptions will be needed.

The following theorem is a generalisation of Theorem 3.4.4 but note that it does not use the stationarity condition.

Theorem 4.1.1. Let U be a random variable, such that $E(U^2) < \infty$, and $\boldsymbol{W}$ an n-dimensional random vector with a finite variance-covariance matrix,

$$\Gamma := Cov(\boldsymbol{W}, \boldsymbol{W}).$$

In terms of mean square error the best linear predictor, $Pred(U|\boldsymbol{W})$, of U based on $\boldsymbol{W} = (W_1, \ldots, W_n)$ satisfies:

(i) $Pred(U|\boldsymbol{W}) = E(U) + \boldsymbol{a}^T (\boldsymbol{W} - E(\boldsymbol{W}))$ where $\Gamma \boldsymbol{a} = Cov(U, \boldsymbol{W})$.

(ii) $E([U - Pred(U|\boldsymbol{W})]) = 0$ i.e. the estimator is unbiased.

(iii) $E([U - Pred(U|\boldsymbol{W})]\, \boldsymbol{W}) = \boldsymbol{0}$, the zero vector.

(iv) $Pred(\alpha_0 + \alpha_1 U_1 + \alpha_2 U_2|\boldsymbol{W}) = \alpha_0 + \alpha_1 Pred(U_1|\boldsymbol{W}) + \alpha_2 Pred(U_2|\boldsymbol{W})$, i.e. the predictor is linear.

(v) $Pred(W_i|\boldsymbol{W}) = W_i$.

Proof. (i) Follows by the same arguments as Theorem 3.4.4 – just the notation changes. While the rest of these results follow directly from (i). □

4.2 Non-stationary time series

The following set of examples illustrates the range of useful linear transformations of stationary processes which can be used in modelling real data.

Example 4.2.1. Consider the random walk with drift, defined by initial condition X_0 and for $t \geq 1$

$$X_t := X_0 + \sum_{i=1}^{t} Z_i,$$

where $Z_t \overset{i.i.d.}{\sim} N(\mu, \sigma^2)$, then $E(X_t) = \mu t$ and $Var(X_t) = t\sigma^2$. If $\mu \neq 0$ the mean is dependent on t – we say it is not *mean stationary*. Further, the variance depends on t so its not *variance stationary*.

We first note that $\{X_t\}$ is a linear function of the stationary process $\{Z_t\}$ and X_0, and that for $t \geq 1$

$$Z_t = \nabla X_t = X_t - X_{t-1} = (1 - B)X_t.$$

So there is a linear transformation, differencing, which was defined in Definition 3.7.1, which recovers the stationary process. Here we have,

$$\nabla X_t \sim \mu + ARMA(0, 0), \tag{4.2}$$

a stationary process with a mean that needs to be estimated.

Example 4.2.2. Let $Z_t \sim WN(0, \sigma^2)$ and define $X_t = \alpha_0 + \alpha_1 t + Z_t$, using a linear transformation. We have that $E(X_t) = \alpha_0 + \alpha_1 t$, so $\{X_t\}$ is not mean stationary. Applying the differencing operator gives

$$\nabla X_t = \alpha_1 + (Z_t - Z_{t-1}).$$

Now $(Z_t - Z_{t-1})$ is an $MA(1)$ process with $\theta_1 = -1$, and is stationary, hence we have

$$\nabla X_t \sim \alpha_1 + ARMA(0, 1), \tag{4.3}$$

where a mean parameter needs to be estimated.

4.2.1 ARIMA modelling

Examples 4.2.1 and 4.2.2 show that non-stationary processes can be closely linked to stationary ones, with differencing being the key tool. We can define a broad class of such examples.

Definition 4.2.3. (d^{th}-order differencing) The operator ∇^d is called the d^{th}-order differencing operator and is defined by

$$\nabla^d X_t = (1 - B)^d X_t.$$

For example,

$$\nabla^2 X_t = (1 - B)(X_t - X_{t-1}) = X_t - 2X_{t-1} + X_{t-2}.$$

Definition 4.2.4. (ARIMA model) If d is a non-negative integer, then $\{X_t\}$ is an *ARIMA*(p, d, q) *process* if

$$Y_t := \nabla^d X_t$$

is a causal ARMA(p, q) process.

4.3 Forecasting ARIMA models

If a non-stationary time series is close to being stationary – in the sense that there exists simple linear transformations which reduce the process to a stationary one and with only a small number of parameters – then we might hope that optimal h-step ahead would be possible.

Example (4.2.1 revisited). Suppose we wish to compute the best linear prediction of X_{n+1} based on $X_0, X_1, \ldots, X_n$ i.e., $Pred(X_{n+1}|X_0, X_1, \ldots, X_n)$. Now, by Theorem 4.1.1 and since

$$X_{n+1} := X_0 + \sum_{i=1}^{n+1} Z_i,$$

we have

$$\begin{aligned} Pred(X_{n+1}|X_0, X_1, \ldots, X_n) &= Pred(X_0 + \sum_{i=1}^{n+1} Z_i|X_0, X_1, \ldots, X_n) \\ &= Pred(X_n + Z_{n+1}|X_0, X_1, \ldots, X_n) \\ &= X_n + Pred(Z_{n+1}|X_0, X_1, \ldots, X_n) \end{aligned}$$

Since $X_0, X_1, \ldots, X_n$ are an invertible linear function of $X_0, Z_1, \ldots, Z_n$ they will have the same best linear predictors, so we have

$$Pred(X_{n+1}|X_0, X_1, \ldots, X_n) = X_n + Pred(Z_{n+1}|X_0, Z_1, \ldots, Z_n) \quad (4.4)$$

To compute this with the same tools we have seen in Chapter 3 we have to make an additional assumption: Y_t is uncorrelated with X_0. This means we have

$$Pred(X_{n+1}|X_0, X_1, \ldots, X_n) = X_n + Pred(Z_{n+1}|Z_1, \ldots, Z_n), \quad (4.5)$$

where the second term is the best linear predictor for a ARMA$(0, 0)$ process with an unknown mean.

Example 4.3.1. (ARIMA$(1, 2, 1)$ forecast) Suppose that we have the model

$$X_t \sim ARIMA(1, 2, 1),$$

and we want to forecast X_{n+2} based on $X_{-1}, X_0, X_1, \ldots, X_n$. By definition we have that $Z_t := \nabla^2 X_t = X_t - 2X_{t-1} + X_{t-2}$ is a causal ARMA$(1, 1)$ process. So

$$X_{n+2} = Z_{n+2} + 2X_{n+1} - X_n,$$

and

$$\begin{aligned} P(X_{t+2}|X_{-1}, X_0, X_1, \ldots, X_n) &= P(Z_{n+2}|X_{-1}, X_0, X_1, \ldots, X_n) \\ &\quad +2P(X_{n+1}|X_{-1}, X_0, X_1, \ldots, X_n) - X_n \end{aligned}$$

We treat each of these terms in turn.

1. First we have, since $Z_1, \ldots, Z_n$ is a linear function of $X_1, \ldots, X_n$, that

   $$P(Z_{n+2}|X_{-1}, X_0, X_1, \ldots, X_n) = P(Z_{n+2}|X_{-1}, X_0, Z_1, \ldots, Z_n),$$

   and if we make the *new* assumption that the Z variables are uncorrelated with X_{-1}, X_0 then we have

   $$P(Z_{n+2}|X_{-1}, X_0, Z_1, \ldots, Z_n) = P(Z_{n+2}|Z_1, \ldots, Z_n)$$

   which is the $h = 2$ forecast of a causal ARMA$(1, 1)$ process which was studied in §3.9 via Theorem 3.4.4.

2. The term $P(X_{n+1}|X_{-1}, X_0, X_1, \ldots, X_n)$, using the same argument as above, can be written as

$$P(Z_{n+1}|Z_1, \ldots, Z_n) + 2X_n - X_{n-1},$$

and hence, again is computable using the theory of stationary processes in Chapter 3

Thus by recursively estimating previous terms we can build up an estimate of X_{n+2} as long as we include extra conditions on the lack of correlation between initial conditions and the driving stationary process.

The ideas behind Example 4.3.1 directly extends to produce h-step ahead forecasts for any ARIMA(p, d, q) models, where we work recursively building the h-step forecast from $(h-1)$-step forecast, etc. Each time we do need to add assumptions on the lack of correlation between initial conditions and the driving causal $ARMA(p, q)$-process. For full details see (Brockwell and Davis, 2002, Page 198).

4.4 Using R for ARIMA modelling

We have already seen, in §3.11, how to use R functions to simulate, fit and predict from ARMA(p, q) processes and almost the exact same methods work for a general ARIMA(p, d, q) process. There are the same two types of problem: (i) how to work with the models for which p, d, q are known, and (ii) how to find appropriate values for p, d, q.

Fitting ARIMA models

Example 4.4.1. (ARIMA model in R) Suppose we have a time series in the `ts`-object `sim.arima.forecast` and that this has been correctly identified as being an ARIMA$(0, 1, 1)$ model – that is of the form Example 4.2.2.

We plot the data in Fig. 4.1. The top left panel shows the time series and the right panel shows the plot of the first difference, by using the `diff( )` function.

We see that the original data does not 'look' stationary – its not centred around a time independent mean with a fixed variance. If we use the `acf( )` function – which really we shouldn't as its only defined for stationary series – we get a plot where the correlations only very slowly decay to zero. On the other hand the auto-correlation of the first difference function looks like that of a stationary series.

If we know the structure of the model we can fit it using the `ARIMA( )` function:

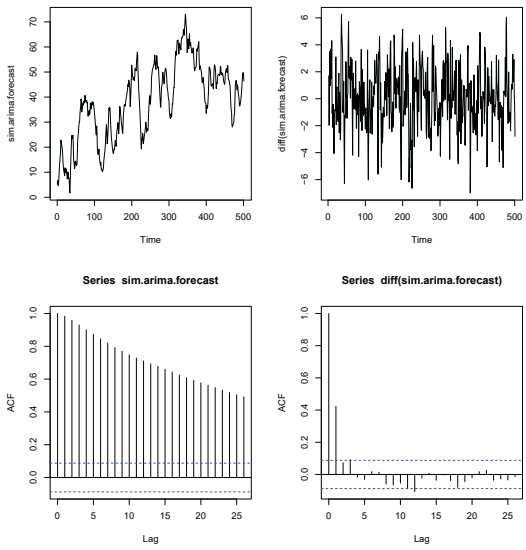

Figure 4.1: Simulation of ARIMA$(0, 1, 1)$ model: top left panel simulated data, top right first difference process, bottom left panel: auto-covariance function of series, bottom right panel auto-covariance of first difference

```
>  arima(sim.arima.forecast, order=c(0,1,1))
Coefficients:
         ma1
      0.5407
s.e.  0.0414
sigma^2 estimated as 4.095:  log likelihood = -1062.07,  aic = 2128.14
```

getting the parameter estimates, standard errors, log-likelihood and AIC values with an assumption of Gaussian errors. The output from this function can go into the `predict( )` function in the following way.

```
> predict(arima(sim.arima.forecast, order=c(0,1,1)), n.ahead=2)
$pred
Time Series:
Start = 502
End = 503
Frequency = 1
[1] 44.97003 44.97003

$se
Time Series:
Start = 502
End = 503
```

```
Frequency = 1
[1] 2.023546 3.71677
```

Figure 4.2 shows the prediction of $h = 20$ steps ahead for this model. We see, unlike the stationary time series example, that the uncertainty in the forecast grows very large for large h. This is a result of the fact that we are trying to forecast a non-stationary process.

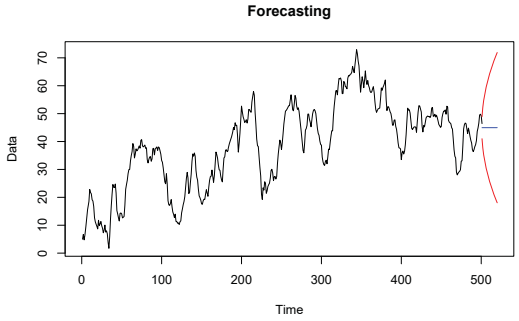

Figure 4.2: Forecasting using the R `predict( )` function. Blue is point forecast and red is ± 1.96 standard errors

Estimating the ARIMA structure

All of the above analysis required us to know the model structure. In general this it not easy to identify model structure exactly but often the search can be reduced to a small number of plausible models. One key trick with a non-stationary time series is to difference it a small number of times and look to see if you can find an acceptable stationary ARMA(p, q) model for the differenced process.

Figure 4.3 shows a concrete example. The top row shows the auto-correlations for an undifference, differenced once and differenced twice simulated series, with the bottom row showing the partial acfs. We see the very slow decay of the auto-correlation which is typical of some non-stationary series. After two differences we see acf and pacf plots which are consistent with stationarity. We could interpret the acf-plot as exponential decay and the pacf-plot is zero after lag 3, so $p = 3, q = 0$ is possible. However so is $p = 1, q = 1$ and $p = 0, q = 2$. In this case the data was generated by a $(1, 2, 1)$ model.

As an alternative to inspecting plots we could use the `AIC` value for model selection as in Chapter 2.

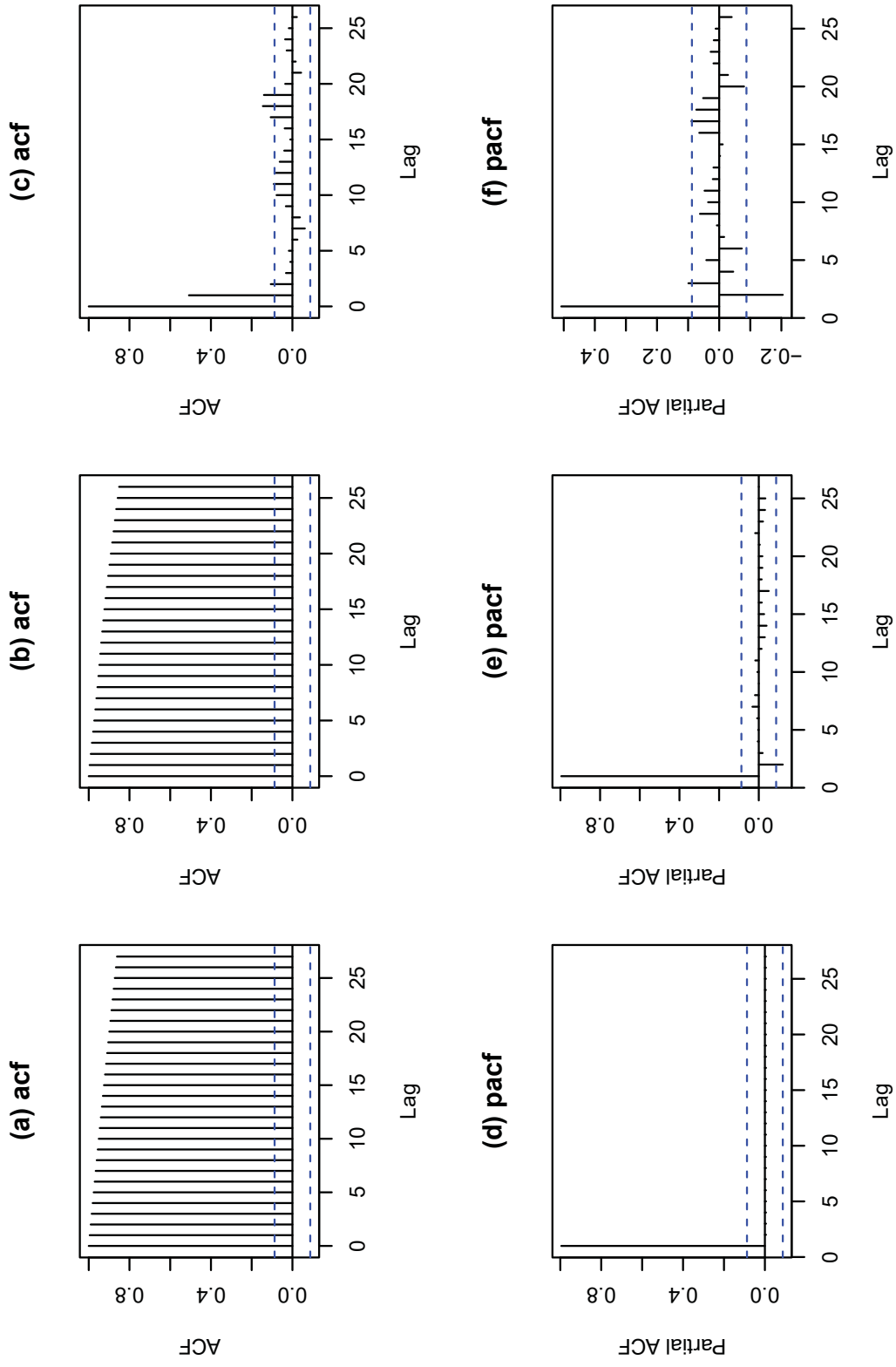

Figure 4.3: No differencing (a, d), once (b,e) and twice (c,f)

4.5 SARIMA modelling

There are other forms of non-stationarity that the Box-Jenkins approach can tackle. The most important is seasonality.

Definition 4.5.1. (Seasonal components) Let $Z_t \sim WN(0, \sigma^2)$ and, for ease of interpretation, let us assume t is recorded on a yearly time scale, but we have one observation per month. Following R we would record January 2015 as 2015.00, February 2015 as $2015.083 = 2015\frac{1}{12}$ etc. Let α_i $i = 1, 2, \ldots, 12$ be constants associate with the month $(i-1)/12$ such that $\sum_{i=1}^{12} \alpha_i = 0$. and define the function $mon(t)$ as the function which returns the month associated with time t.

The process $X_t := \alpha_{mon(t)} + Z_t$ where $\{Z_t\} \sim ARIMA(p, d, q)$ is said to have a seasonal component. It will not be stationary if any of the α_i values are nonzero.

Example 4.5.2. (US accident data)

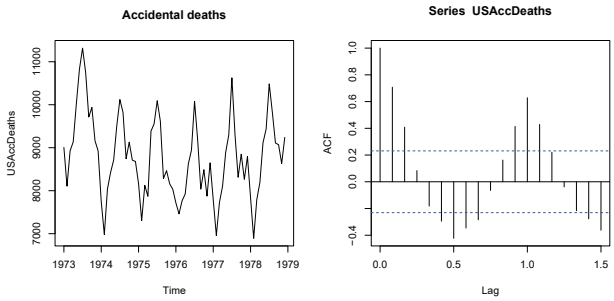

Figure 4.4: US accidental deaths data 1973- 1978. Data and auto-correlation plot.

Definition 4.5.3. (Lag s difference) The lag s difference operator B^s is defined by

$$B^s X_t = X_{t-s}.$$

The corresponding difference operator Δ^s is then defined as

$$\Delta^s := (I - B^s).$$

Differencing at appropriate lags can remove seasonality, hence are another example where a linear function of a non-stationary time series can be stationary. This gives rise, in a very similar way to the definition of the $ARIMA$ model, to a wider class of models which can model seasonal time series.

Definition 4.5.4. (Seasonal $ARIMA$ (SARIMA) model) If d and D are nonnegative integers, then $\{X_t\}$ is a *seasonal* $ARIMA(p,d,q)(P,D,Q)$ *process* with period s if the differenced series

$$Y_t := (1-B)^d(1-B^s)^D X_t,$$

is a causal ARMA process defined by

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_t,$$

where $Z_t \sim WN(0,\sigma^2)$ and

$$\begin{aligned}
\phi(z) &:= 1-\phi_1 z-\cdots-\phi_p z^p \\
\Phi(z) &:= 1-\Phi_1 z-\cdots-\Phi_P z^P \\
\theta(z) &:= 1+\theta_1 z+\cdots+\theta_q z^q \\
\Theta(z) &:= 1+\Theta_1 z+\cdots+\Theta_Q z^Q
\end{aligned}$$

The corresponding process is causal if and only if all the roots of $\phi(z)$ and $\Phi(z)$ lie outside the unit circle.

4.6 Using R for SARIMA modelling

Just as in §4.4 there are the same two types of problem: (i) how to work with the models for which p,d,q,P,D,Q are known, and (ii) how to find appropriate values for p,d,qP,D,Q.

Fitting SARIMA models

Example 4.6.1. (SARIMA model in R) Let us look at a particular example for concreteness. The R data object `USAccDeaths` contains a monthly time series of accidental deaths in the USA in the period 1973 -1978. Figure 4.5 (a) shows the data and we see a strong seasonal effect. Differencing at lag 12 is shown in Panel (b) and there seems to be a trend so another difference is done to give Panel (c) which 'looks stationary'. The R code which did this is as follows.

```
> data(USAccDeaths)
> plot(USAccDeaths, main="(a) Accidental deaths", ylab="Counts")
> plot(diff(USAccDeaths, lag=12) )
> plot(diff(diff(USAccDeaths, lag=12)))
```

So it looks like a SARIMA model might be appropriate. Let us suppose that we know that $(p,d,q)=(1,1,1)$ and $(P,D,Q)=(0,1,1)$ we then fit the appropriate SARIMA model using:

```
>  mod2 <- arima(USAccDeaths, order = c(1, 1,1),
      seasonal = list(order = c(0, 1, 1), period = 12)
```

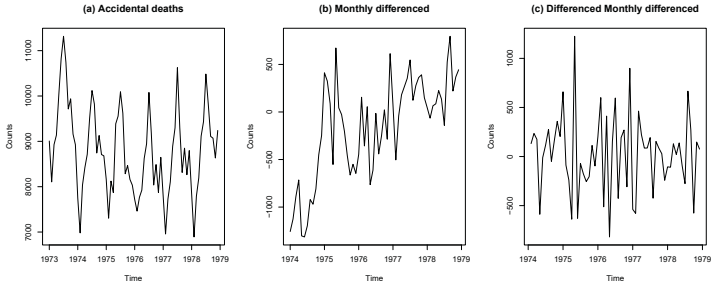

Figure 4.5: US accident data

where `mod2` is an `arima` object containing the estimated parameters and other details of the fit i.e.

```
> mod2
Call:
arima(x = USAccDeaths, order = c(1, 1, 1),
   seasonal = list(order = c(0, 1, 1),  period = 12))
Coefficients:
         ar1      ma1     sma1
      0.0979  -0.5109  -0.5437
s.e.  0.3111   0.2736   0.1784
sigma^2 estimated as 99453:  log likelihood = -425.39,  aic = 858.78
```

We need to see if the fit is 'good' so we look the residuals using

```
> acf(mod2$residuals, main="ACF residuals")
> acf(mod2$residuals, type="partial", main="PACF residuals")
```

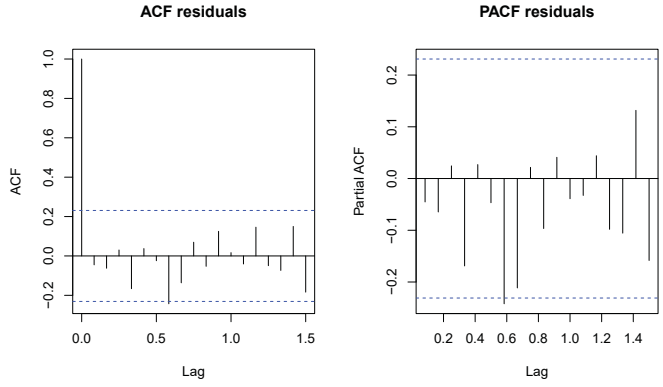

Figure 4.6: US accident data

We can then make predictions for two years ahead using

```
> predict(mod2, n.ahead=24)
```

and the results can be found in Fig. 4.7 with the red line the point forecast and the blue the prediction intervals.

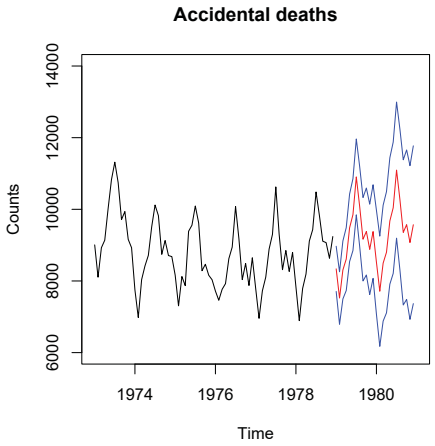

Figure 4.7: US accident data

Estimating the SARIMA structure

As in the case of ARIMA modelling we can inspect acf and pacf plot to look for possible p, d, q and P, D, Q values. This usually requires a good deal of experience to do reliably and a good deal of data.

Example (4.6.1 revisited). We have already seen by looking at differencing plots that $d = 1, D = 1$ are plausible – although not the only possible values. We will stick to those and look a acf and pacf plots of these differences in Fig. 4.8 To try and guess P, Q we look for the same types of patterns discussed above –i.e. exponential decay and finding a bound after which all estimates are zero – but only for integer values. That is we look at the yearly lags at $1, 2, \ldots$. Here we see $P = 0, Q = 1$ is possible, although not the only possible choice. To think about the p, q values we look for patterns at the monthly $\frac{1}{12}, \frac{2}{12}, \ldots$ lags. Here we see that $p = 1, q = 1$ is possible, as are other choices.

An alternative method to interpreting this plots is look look at all models with small p, q, P, Q values and look at the AIC value given by the `arima( )` function output.

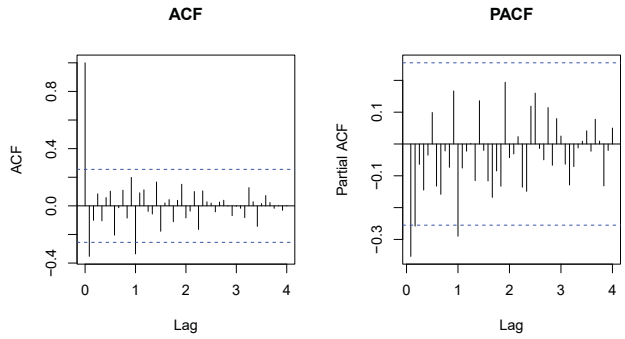

Figure 4.8: US accident data

4.7 The Box Jenkins Approach

4.7.1 Summary

In this chapter we have looked at the Box-Jenkins methodology for forecasting. It is based on the idea that many stationary processes can be parsimoniously represented by $ARMA(p, q)$ models – that is only a relatively small number of parameters are need. As we have seen in Chapter 2 if we can fit observed data with only a small number of parameters then often we can get reasonable forecasting models.

However, most applications involve some form of non-stationarity and the methodology deals with this by appropriate differencing combined with associated independence assumptions. We can summarise the steps as follows

(a) Model Identification. This uses the R functions `plot, diff, acf` etc.

(b) Estimation. Once the structure of the model is identified we used the `arima` function to estimate the parameters

(c) Diagnostic checking. Using the residuals from the model fit we can check to see if the model is 'satisfactory'.

(d) Consider alternative models if necessary.

4.7.2 Tests for stationarity

One of the important stages of the approach is to decide if a suitably difference time series is stationary. This can be rather subjective if purely graphical methods are used. We can use more formal tests which focus on different kinds ways that stationarity can fail.

Definition 4.7.1. (Phillips-Perron test) The Phillips-Perron test is a test that the time series has a 'unit root' a stationary alternative. Recall that we check for stationarity in an ARMA process by looking to see where the roots of the polynomial $\phi(z)$ lie. If they lie on the unit circle then we say the series has a unit root and is non-stationary. The random walk model is the typically example of this.

In R we use the function `PP.test`. For the data in Example 4.6.1 selecting $d = D = 1$ reduced the model to stationarity. If we try and test this formally we get

```
> PP.test(diff(diff(USAccDeaths, lag=12)))

Phillips-Perron Unit Root Test

data:  diff(diff(USAccDeaths, lag = 12))
Dickey-Fuller = -11.8917, Truncation lag parameter = 3, p-value = 0.01
```

Definition 4.7.2. (Runs test) If there was a trend in the sequence the series would have a pattern in if it was above or below its mean. The runs test looks at the the frequency of runs of a binary data set. The null hypothesis is that the series is random.

```
> library(tseries)
> x <- diff(diff(USAccDeaths, lag=12))
> y <- factor(sign(x - mean(x)))
> runs.test(y)

Runs Test

data:  y
Standard Normal = 0.2439, p-value = 0.8073
alternative hypothesis: two.sided
```

Here we convert the continuous time series to a binary one and then use the `runs.test( )` from the `tseries` library.

4.7.3 Criticisms of approach

The paper Chatfield and Prothero (1973) has some criticisms of the Box-Jenkins approach. In particular we note the following.

(a) Differencing to attain stationarity may introduce spurious auto-correlations. For example with monthly data it can occur at lag 11.

(b) It can be difficult, just from data, to distinguish between different SARIMA models and these may give different forecasts

(c) A reasonably large number of data points is needed to implement the approach with Box and Jenkins (1976) stating that $50-100$ observations are needed.

(d) ARIMA models may be harder for practitioners to understand than linear trends plus seasonal effect models.

5

Bayesian and state space methods

5.1 Introduction

Most of the standard results in this section can be found in a number of places including Berger (2013) or Jackman (2009) and we use a number of R-packages including `dlm`, and `rjags` which links to the *JAGS* program, Plummer (2010), which is not part of R but can be run from it.

5.2 Bayesian methods

Finally we briefly review the ideas of Bayesian inference. We shall assume that we have a model for the data $f(x|\theta)$ and that we want to learn about the (vector) of parameters θ. In Bayesian statistics everything is considered a random variable and all statements about uncertainty are made using the language and calculus of probability theory.

Suppose, that we consider that the parameter vector θ itself is random, and that it has a distribution, $Pr(\theta)$ which we shall call the *prior distribution*. This represents what we know about θ *before* we see the data. In the Bayesian viewpoint all uncertainty – and in particular what we know about the value of θ – can be represented by probability statements.

To see what we know about θ *after* we see the data we apply Bayes theorem to give

$$Pr(\theta|\text{ data }) = \frac{Pr(\text{ data }|\theta)Pr(\theta)}{Pr(\text{data})}. \tag{5.1}$$

The term $Pr(\theta|\text{ data })$ is called the *posterior* and is interpreted as what we know about θ after we have seen the data. We can the write 5.1 as

$$\text{Posterior}(\theta) \propto Lik(\theta) \times \text{prior}(\theta) \tag{5.2}$$

That is, up to a constant, the likelihood function is the way we convert the information we have about θ before we have seen the data into the information we have about it after seeing the data.

Example 5.2.1. (Normal Bayesian inference 1) Consider a simple normal example, where we have a model $X_i \overset{i.i.d.}{\sim} N(\theta, 1)$ with a sample size of n, so that the likelihood function based on the observed data $x_1, \ldots, x_n$ is

$$\begin{aligned} Lik(\theta) &= Pr(data|\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(x_i-\theta)^2}{2}\right] \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left[-\sum_{i=1}^{n} \frac{(x_i-\theta)^2}{2}\right] \\ &\propto \exp\left[\theta(\sum_{i=1}^{n} x_i) - \frac{n}{2}\theta^2\right] \end{aligned}$$

This is then the key tool that we need to merge with the prior in order to get the posterior.

In the Bayesian viewpoint the 'answer' to an inference problem is captured in the posterior distribution. There are then a number of computational issues which we need to study. First we note that the very simple formula Equation (5.2) only defines the posterior up to a constant. Since the posterior is a probability mass function or a density function we might want to be able to deal with the fact that finding

$$\int_{\theta} \text{Posterior}(\theta)\, d\theta,$$

might be a difficult computational problem. This can be solved either by clever mathematics or by numerical integration methods. Secondly, we want to know how can we convert mathematical knowledge about the posterior into something interpretable? Again this looks like this will involve (potientially) difficult integration problems.

Example (5.2.1 revisited). Suppose that the prior distribution for θ was in the Normal family i.e.

$$\theta \sim N(\theta; \mu_{\text{prior}}, \sigma^2_{\text{prior}}) \propto \exp\left[-\frac{(\theta-\mu_{\text{prior}})^2}{2\sigma^2_{\text{prior}}}\right]$$

where μ_{prior} and σ^2_{prior} are considered known constants. Then the posterior

can be written as

$$\begin{aligned} Pr(\theta|x_1,\ldots,x_n) &\propto Lik(\theta)\times \text{prior}(\theta) \\ &\propto \exp\left[\theta(\sum_{i=1}^{n} x_i) - \frac{n}{2}\theta^2\right] \times \exp\left[-\frac{(\theta-\mu_{\text{prior}})^2}{2\sigma^2_{\text{prior}}}\right] \\ &\propto \exp\left[-\frac{1}{2}\frac{\left(1+n\sigma^2_{\text{prior}}\right)\theta^2 - 2\left(\mu_{\text{prior}}+\sigma^2_{\text{prior}}(\sum_{i=1}^{n} x_i)\right)\theta}{\sigma^2_{\text{prior}}}\right] \\ &= \exp\left[-\frac{1}{2}\frac{\theta^2 - 2\frac{\left(\mu_{\text{prior}}+\sigma^2_{\text{prior}}(\sum_{i=1}^{n} x_i)\right)}{\left(1+n\sigma^2_{\text{prior}}\right)}\theta}{\sigma^2_{\text{prior}}/\left(1+n\sigma^2_{\text{prior}}\right)}\right] \end{aligned}$$

So that we have

$$Pr(\theta|x_1,\ldots,x_n) \sim N\left(\frac{\left(\mu_{\text{prior}}+\sigma^2_{\text{prior}}(\sum_{i=1}^{n} x_i)\right)}{\left(1+n\sigma^2_{\text{prior}}\right)}, \frac{\sigma^2_{\text{prior}}}{\left(1+n\sigma^2_{\text{prior}}\right)}\right) \quad (5.3)$$

So, when we see data, the prior distribution $N(\theta;\mu_{\text{prior}},\sigma^2_{\text{prior}})$ is updated to another normal but with different mean and variance. As n gets larger the mean of the posterior satisfies

$$\frac{\left(\mu_{\text{prior}}+\sigma^2_{\text{prior}}(\sum_{i=1}^{n} x_i)\right)}{\left(1+n\sigma^2_{\text{prior}}\right)} \approx \bar{x} \to \mu,$$

the mean of the model. The variance will satisfy

$$\frac{\sigma^2_{\text{prior}}}{\left(1+n\sigma^2_{\text{prior}}\right)} \approx \frac{1}{n} \to 0.$$

We interpret this as saying, when we have a lot of data from the model the posterior converges to a spike distribution around the true value.

In general with Bayesian methods, the nice properties of Example 5.2.1 hold in general. As we get more data we update the posterior and it will shrink around the 'true' values of the model.

One property that is special is the that the prior distribution (a normal here) was updated to another normal. In general the posterior will not be a simple well-known distribution and the normalising factor has to be computed numerically.

Having seen how to compute the posterior distribution we now see how the Bayesian method deals with forecasting. The same principles apply: only use the rules of probability.

Definition 5.2.2. (Predictive distribution) The solution to a forecasting problem to forecast X_{new} based on an observed sample $X_1, \ldots, X_n$ which comes from a model $f(X|\theta)$ should be the conditional distribution $P(X_{\text{new}}|X_1, \ldots, X_n)$. This is called the *predictive distribution* and is calculated from the posterior by

$$P(X_{\text{new}}|X_1, \ldots, X_n) = \int_\Theta P(X_{\text{new}}|\theta)P(\theta|X_1, \ldots, X_n)d\theta.$$

In words we average over the model according to the posterior belief in the value of θ given the observed data.

Example (5.2.1 revisited). For our example we have the predictive distribution

$$\int \phi(x; \theta, 1)\phi\left(\theta; \mu_{\text{Post}}, \sigma^2_{\text{Post}}\right) d\theta.$$

where $\phi()$ is the normal density function. After some algebra this can be shown to be

$$\phi(x; \mu_{\text{Post}}, 1 + \sigma^2_{\text{Post}}) \tag{5.4}$$

Definition 5.2.3. (Inverse gamma distribution) A positive random variable $X \sim InvGamma$ has an inverse gamma distribution if X^{-1} has a gamma distribution. Its density is

$$f_X(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\{-\beta/x\}$$

for $x > 0, \alpha > 0$ and $\beta > 0$. Here α is called the shape parameter and β the scale.

Example 5.2.4. (Normal Bayesian inference 2) Suppose now we have the model $X_i \overset{i.i.d.}{\sim} N(\theta, \sigma^2)$ with a sample size of n where both θ and σ^2 are to be estimated. We can take the priors

$$\begin{aligned} \theta|\sigma^2 &\sim N\left(\mu_0, \frac{\sigma^2}{n_0}\right) \\ \sigma^2 &\sim InvGamma(\nu_0/2, \nu_0\sigma_0^2/2) \end{aligned}$$

In this prior specification n_0 is a known constant called the 'prior sample size' – a nice way of measuring the relative weight of the prior and the data. The terms ν_0 and σ_0^2 are considered known.

For this set up, after some algebra, we get the posterior for μ, σ^2 as

$$\begin{aligned} \theta|\sigma^2 &\sim N\left(\mu_1, \frac{\sigma^2}{n_1}\right) \\ \sigma^2 &\sim InvGamma(\nu_1/2, \nu_1\sigma_1^2/2) \end{aligned}$$

where

$$\mu_1 := \frac{n_0\mu_{\text{prior}} + n\bar{x}}{n_0 + n}, n_1 = n_0 + n, \nu_1 = \nu_0 + n$$

and

$$\nu_1\sigma_1^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2 + \nu_0\sigma_0^2 + \frac{n_0 n}{n_1}(\mu_0 - \bar{x})^2$$

In Example 5.2.4 the priors have been selected in a clever way so that the function form does not change before and after see the data. These are called *conjugate priors*. They are very convenient, keep the mathematics very clean, but not strictly necessary.

5.2.1 Bayesian methods in regression

We can generalise the results of Example 5.2.4 to the standard regression framework we studied in Chapter 2. As Jackman (2009) shows we can use the same conjugate prior 'trick' and keep an analytic solution possible.

Definition 5.2.5. (Conjugate prior) A *conjugate prior* is a prior which, for a given model, has the property that the posterior lies in the same functional family as the prior. The updating using the data only changes the parameter values in the conjugate family.

Computationally we don't have to do that and we can, using R compute much more general cases using Markov chain Monte Carlo (MCMC) analysis.

Definition 5.2.6. (MCMC) If we have a complex distribution $P(\theta)$ that is known up to a constant then rather than working directly with the distribution we can compute most statistical functions – such as means, variances, marginal distributions – using a large sample $\theta_1, \theta_2, \ldots$ which is a Markov chain whose stationary (equilibrium) distribution is proportional to $P(\theta)$.

We will not, in this course, look at the details of how the sample $\theta_1, \theta_2, \ldots$ is constructed but rather use built in R functions.

Example 5.2.7. (Bayesian regression) Suppose we have a model

$$y_i|\boldsymbol{x}_i \sim N(\boldsymbol{x}_i\beta, \sigma^2)$$

with semi-conjugate independent priors

$$\begin{aligned} \beta &\sim MN_{p+1}\left(b_0, B_0^{-1}\right) \\ \sigma^2 &\sim InvGamma(c_0/2, \nu_0, d0^2/2) \end{aligned}$$

then the posterior is could be calculated but it is easier to use MCMC.

Example (5.2.4 revisited). We can think of this example as a regression example with no covariates. In R we could use

```
> library(MCMCpack)  #load library
> library(coda)
> y <- rnorm(30, mean=3, sd=0.2) #generate data
> out <- MCMCregress(y~ 1)  #run mcmc
> out1 <- as.matrix(out)   #store output as matrix
> dim(out1)
[1] 10000     2
```

The output is a set of two long series, one for θ and one from σ^2. These large samples represent the (joint) posterior of both parameters and are plotted as density estimates in Fig. 5.1 (b) and (c). Panel (a) is just a plot of the data used.

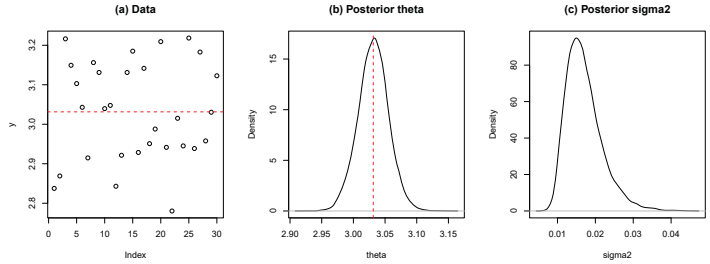

Figure 5.1: MCMC output for Example 5.2.4

We can look at more detail at this output. In Fig. 5.2, Panel (a) shows the first 500 points of the θ-series, noting that there are in fact $10,000$ points in the series. Panel (b) shows the same for the σ^2 series and (c) shows a joint plot. This last shows out the MCMC outputs from the joint distribution of the posterior.

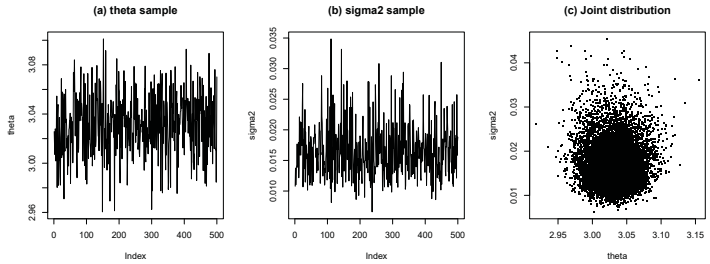

Figure 5.2: MCMC output for Example 5.2.4 (a) θ sample, (b) σ^2 sample, (c) joint distribution

If we want to compute posterior means, standard deviations etc we simply use the sample i.e.

```
> mean(out1[,1])
```

```
[1] 3.031442
> sd(out1[,1])
[1] 0.02407227
```

which give point estimates of the parameters and measures of (posterior) uncertainty.

Panel (c) shows a scatterplot of the output. This is a representation of the *joint* posterior distribution of the parameters. The fact that a Bayesian analysis gives joint inference is very different from the standard methods which give, say, confidence intervals for each parameter one at a time. In this simple random normal sample example since the sampling distribution of the sample mean and sample variance are independent this is rarely an issue. For example panel (c) looks independent so two marginal inferences are sufficient. In general though the dependence can really be important.

Example (2.3.1 revisited). We can return to the house price example of Chapter 2 and fit the model using MCMC.

```
> out <- MCMCregress(Y ~X1+X2+X3+X4+X5+X6+X7+X8+X9, data=house.price)
> summary(out)

                Mean      SD  Naive SE Time-series SE
(Intercept) 15.22298 6.45155 0.0645155      0.0645155
X1           1.93971 1.12235 0.0112235      0.0112235
X2           6.87051 4.67551 0.0467551      0.0480019
X3           0.13595 0.53313 0.0053313      0.0053274
X4           2.80602 4.75816 0.0475816      0.0475816
X5           2.04593 1.49712 0.0149712      0.0149712
X6          -0.50892 2.60459 0.0260459      0.0260459
X7          -1.29140 3.70783 0.0370783      0.0370783
X8          -0.03864 0.07188 0.0007188      0.0007188
X9           1.69108 2.09140 0.0209140      0.0209140
sigma2      10.25968 4.46910 0.0446910      0.0725849
```

We get very similar estimates to the OLS solution. But now, since we are getting a joint posterior for all the parameters, we can see something interesting about the multicollinearity issue seen before. Recall that we found it odd that the estimate for β_6 was negative and this was explained due to dependence in the explanatory variables. In the Bayesian analysis this dependence flows very naturally into the posterior as shown in Fig. 5.3. We see our knowledge about the value of β_6 is strongly tied to our knowledge about other parameters, say β_2 or β_6. This gives a much more complete picture of what we really know about the parameters even when there is multicollinearity.

How do we use the MCMC output to make predictions. We want to use the predictive distribution of Definition 5.2.2. This is an integral and with

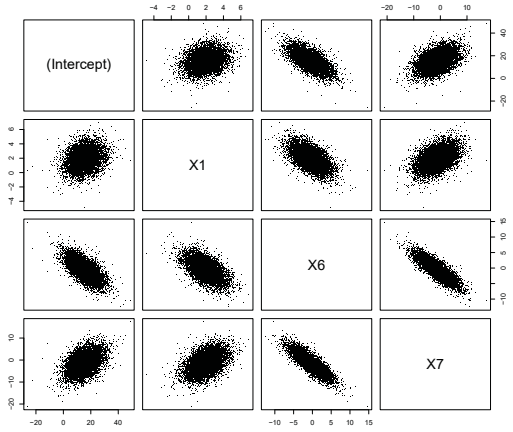

Figure 5.3: MCMC output for Example 2.3.1

MCMC output all integrals – which are hard mathematically – are replaced with sums over the MCMC sample i.e.

$$\int_{\Theta} f(\theta)P(\theta|X_1,\ldots,X_n)d\theta \approx \sum_{i=1}^{N_{MCMC}} f(\theta_i) \tag{5.5}$$

for any function f where θ_i for $i = 1,\ldots,N_{MCMC}$ is the MCMC sample.

Definition 5.2.8. (Building a predictive sample) If we want to do prediction using a posterior for θ we use a similar idea. We can take a sample from the predictive distribution in two steps:

(i) Draw a sample $\theta^{(i)}$ from the posterior $P(\theta|data)$

(ii) Given $\theta^{(i)}$ draw a prediction from the model $f(y|\theta^{(i)})$

We repeat this, independently for many different values of i.

Example (2.3.1 revisited). In the regression example, given $\beta^{(i)}$ we take the simple forecast $y^{(i)} = x_0^T\beta^{(i)} + \epsilon^{(i)}$, where $\epsilon^{(i)} \sim N(0, \sigma^{2\,(i)})$. We could look at the distribution of this over values of β from the posterior. In Figure 5.4 we select x_o to be the average house and generate a sample from the predictive distribution using the two steps of Definition 5.2.8. The blue vertical line is the sample mean of the house prices which, because of the choice of x_0, should be the mean here.

5.2.2 Link to penalty methods

From Hastie et al. (2009, p. 64) we get the following link between the ridge regression method of Definition 2.4.7 and Bayesian methods. They show

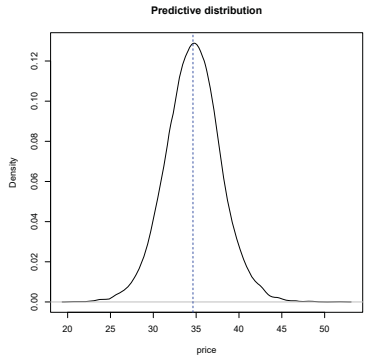

Figure 5.4: Predictive distribution for Example 2.3.1

that ridge regression estimates are modes of posterior distributions with suitably chosen priors.

Theorem 5.2.9. (Bayes and ridge regression) Suppose we have the model for Y conditionally on X as

$$Y_i|X, \beta \sim N(\beta_0 + x_i^T \beta, \sigma^2)$$

with independent priors $\beta_j \sim N(0, \tau^2)$. Then, with τ, σ assumed known the log-posterior for β is given

$$\sum_{i=1}^{N}(y_i - \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

which agrees with $RSS(\lambda)$ in Equation 2.5 when $\lambda = \sigma^2/\tau^2$.

The LASSO can also be interpreted in Bayesian terms.

Theorem 5.2.10. (Bayes and ridge regression) The Lasso estimate for linear regression parameters can be interpreted as a Bayesian posterior mode estimate when the priors on the regression parameters are independent double - exponential (Laplace) distributions.

Proof. See Park and Casella (2008). □

5.3 Dynamic linear modelling

Having looked at some simple Bayesian examples with conjugate priors and using MCMC in regression we now look at Bayesian method with time series models. We will follow the approach of West and Harrison (1997) and Petris et al. (2009) in particular the R library `dlm` from that book, and `rjags` which links to the *JAGS* program, Plummer (2010),

5.3.1 State space model

Definition 5.3.1. (State space model) If $\{Y_t\}$ is an (observed) time series with (unobserved) state process $\{\theta_t\}$. In a state space model the dependence between these is defined by the graphical model shown in Fig. 5.5 The re-

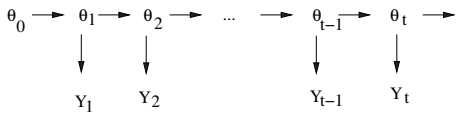

Figure 5.5: State space structure

lationship defines the conditional independence relations: (i) θ_t is a Markov chain (ii) conditionally on θ_t for $t = 0, 1, \ldots, t$ then Y_t are independent so that the joint distribution of $(Y_1, \ldots, Y_t) \,|\, (\theta_1, \ldots, \theta_t)$ is

$$\prod_{i=1}^{t} f(y_t|\theta_t).$$

State space models are also called *hidden Markov models*.

5.3.2 The (normal) dynamic linear model (DLM)

We want a notation which shows what information is available to us in order to make a forecast. In the Bayesian methodology this can be more complex than just the observed time series. Any information which can be expressed in terms of a probability statement can be used and would go into the prior. This can include expert knowledge from the forecaster.

Definition 5.3.2. (Information available) Let the information set available to the forecaster at time t be denoted by D_t. Note that since a Bayesian analysis can, in principle, incorporate expert knowledge – as long as it can be written in terms of a probability distribution – then D_t might be larger than the observed time series $\{y_0, y_1, \ldots, y_t\}$.

To use the Bayesian method we need a probability model for the time series, $f(Y_t|\theta_t, D_{t-1})$ where θ_t is a vector of parameters that may, or may not, depend on t and can have parameters added if circumstances change. If the parameters depend on time then we say we have a *dynamic model*. To complete the Bayesian framework we will also need a prior distribution of $\theta_t|D_{t-1}$.

Following Petris et al. (2009) we will make the following definition.

Definition 5.3.3. (Filtering, smoothing and prediction) In a Bayesian analysis of a DLM we are primary interesting the the posterior $P(\theta_s|y_1, \ldots, y_t)$ and we subdivide the kind of problem by: (i) *filtering* is the case where

(s=t), (ii) *state prediction* is the case when $s > t$ and (iii) *smoothing* is the case $s < t$.

We also have the observation predictive distribution $P(y_{t+k}|y_1, \ldots y_t)$ for $k > 0$.

Definition 5.3.4. (General dynamic linear model) The general form of the the dynamic linear model (DLM) is

$$\begin{aligned} Y_t &= F_t\theta_t + v_t, \\ \theta_t &= G_t\theta_{t-1} + w_t, \end{aligned}$$

where $v_t \sim N(0, V_t)$, $w_t \sim N(0, W_t)$ are independent. Here G_t, W_t are known matrices and V_t and W_t are covariance matrices, which can be treated as being known or unknown. The terms θ_t are unobserved and call the *state* of the system, while Y_t are observed. The initial conditions for the system are $\theta_0 \sim N(m_0, C_0)$

Definition 5.3.5. (Univariate normal DLM) The univariate normal dynamic linear model (DLM) is defined by the notation $DLM\{F_t, \lambda, V_t, W_t\}$ where

$$\begin{aligned} Y_t &= F_t\mu_t + v_t \qquad \text{(Observation equation)} \\ \mu_t &= \lambda\mu_{t-1} + w_t, \qquad \text{(System equation)} \end{aligned}$$

where $v_t \sim N(0, V_t)$, and $w_t \sim N(0, W_t)$, and the initial information is $\mu_0|D_0 \sim N(m_0, C_0)$. We assume that all v_t and w_t are separately and jointly independent. In this model F_t is specified and the variance terms V_t, W_t can be treated either as known or unknown.

Definition 5.3.6. (Constant model) The simplest case of a DLM is denoted by $N(\mu, V)$, in our representation is $DLM\{1, 1, V, 0\}$ which is the constant mean model

$$Y_t = \mu + v_t$$

where $v_t \sim N(0, V)$. Here there are no dynamics for the μ_t terms

Unlike Definition 5.3.6 we might want a model where the mean may change with time. In Chapter 1 we found the idea of a 'slowly varying function' to be useful. The following DLM is one way to formalise this.

Definition 5.3.7. (Simple DLM) The model $DLM\{1, \lambda, V, W\}$ has

$$Y_t = \mu_t + v_t$$

where $v_t \overset{i.i.d.}{\sim} N(0, V)$, so μ_t represents the level of the process at time t and v_t is the error around the underlying level. To let the mean vary is we can define

$$\mu_t = \lambda\mu_{t-1} + w_t$$

where $w_t \overset{i.i.d.}{\sim} N(0, W)$, i.e. an AR(1) process for the mean. If the variance W here is 'small' the mean can only slowing change its value.

Example 5.3.8. (Simulations of DLM) The examples in Fig. 5.6 are four examples of simple DLM. In each $W = 1$ and we vary V – to change the 'signal to noise ratio' – and λ to change the dynamics of the μ_t process.

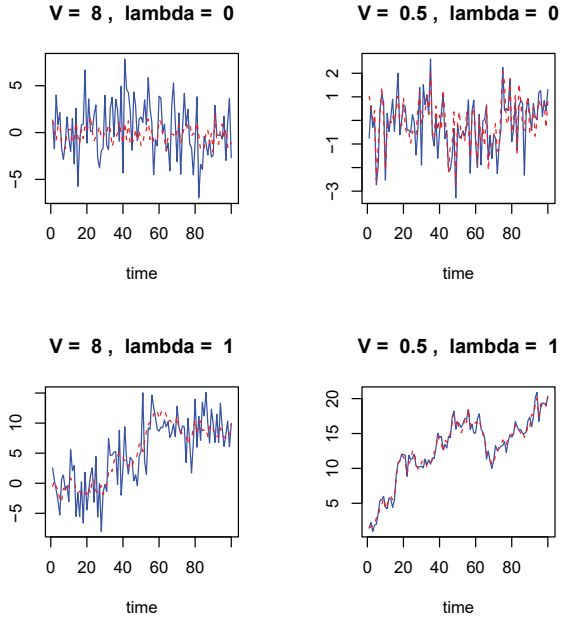

Figure 5.6: Simulations of simple DLMs: red is μ_t and blue is Y_t

Definition 5.3.9. (Local linear trend) A more complex model would include a term, β_t, for the *local linear trend*. This is similar to an idea we have seen in the Holt-winters method; Def. 1.5.6. Here the model is defined as

$$\begin{aligned} Y_t &= \mu_t + \nu_t, \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + \omega_{1,t}, \\ \beta_t &= \beta_{t-1} + \omega_{2,t}, \end{aligned}$$

where

$$G = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, W = \begin{pmatrix} \sigma_{w_1}^2 & 0 \\ 0 & \sigma_{w_2}^2 \end{pmatrix}, F = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

and ν_t, ω_t independent zero mean Normal random variables.

Example 5.3.10. (Simulation of local linear trend model)

Figure 5.7 shows an example of the local linear trend model via simulated data. The blue line is the trend which – in this example is a random walk. This can be positive or negative and as it becomes larger in absolute value

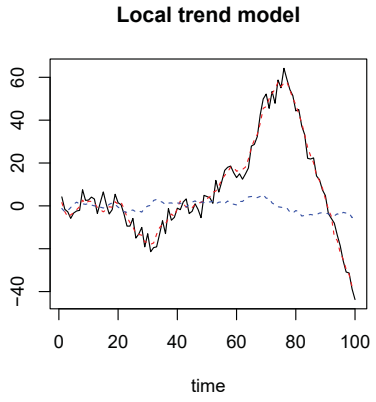

Figure 5.7: Simulations of simple DLMs: red is μ_t and black is Y_t and blue is β_t

it dominates the dynamics. The red line is the term μ_t and the black line the observed time series.

We can also include covariates in these models.

Definition 5.3.11. (Simple dynamic regression) Suppose we have a covariate x_t which has information about the time series of interest. We can include it into our model using the formulation

$$Y_t = \theta_{1,t} + \theta_{2,t} x_t + \epsilon_t$$

where we allow θ to be slowing varying via

$$\theta_t = \lambda \theta_{t-1} + \omega_t.$$

5.4 Working with dlms in R

5.4.1 The `dlm` package

In R we can implement the DLM using the `dlm` package. For example

```
> library(dlm)
>   rw <- dlm(m0=  mean(Nile) , C0= var(Nile),
      FF=1, V= 1.192585e+04,  GG=1, W= 2635.85)
```

loads the library and defines a `dlm` object. In Definition 5.3.4 we see, to specify a DLM we need the terms m_0, C_0 to define the initial conditions, F_t, V_t to define the Y_t given θ_t, G_t, W_t to define the dynamics of θ_t. In the R code these are defined by `m0, C0, FF, V, GG, W`, respectively. We will

discuss how to 'fit' these models in §5.4 but for the moment we treat the parameters as if they are known.

Having defined a model we can fit it to data using the `dlmFilter` function, for example

```
> rwFilt <- dlmFilter(Nile, rw)
> plot(rwFilt$y, xlim=c(1860,1990))
> lines(rwFilt$f, col="red")
```

generates Fig. 5.8.

Example 5.4.1. (Nile data) The black line in Fig. 5.8 shows measurements of the annual flow of the river Nile at Ashwan in the period 1871-1970, Durbin and Koopman (2012) and R Core Team (2013) The red line shows the dlm filter defined above, while the blue lines show prediction intervals for this model. One interesting aspect of this data is that there is a change

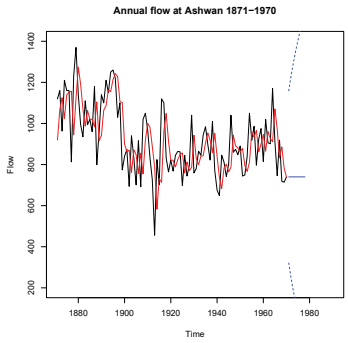

Figure 5.8: Nile data: black line observed data, red line dlm filter, blue lines forecast

point where the Asham high dam was built in the 1960's.

5.4.2 JAGS, BUGS and R

As we have seen above we can use MCMC to compute any posterior quantity in a Bayesian setting. One major issue with MCMC is computational speed and a number of packages have been written to help users efficiently implement MCMC in complex hierarchical models. We will focus on JAGS, which can be run directly from R.

BUGS and WinBugs[1] were highly successful packages that was made available to users which had a major impact on using MCMC methods in many areas. A very good and up to date introduction can be found in Lunn et al. (2012). BUGS stands for 'Bayesian inference Using Gibbs Sampling' and the WinBUGS package was developed to run on Windows machines.

[1] http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

In this course we will concentrate on the package JAGS which is a clone of BUGS (Bayesian analysis Using Gibbs Sampling) which runs on Windows, Mac and UNIX machines and through the R package *rjags* can be easily run directly from R. Note that the *rjags* package does not include a copy of the JAGS library which must be installed separately[2]. JAGS (Just Another Gibbs Sampler) is licensed under the GNU General Public License and is thus freely available.

Let us look at a very simple example of running a Gibbs sampler in run from R using JAGS.

Example 5.4.2. (Independent normal example) Consider the model

$$\begin{aligned} X|\theta,\tau &\sim N(\theta,\tau^{-1}) \\ \theta|\tau &\sim N(0,100/\tau) \\ 1/\tau &\sim Gamma(1/1000,1/1000) \end{aligned}$$

First, in a text file – which I have called *jagsmodel2.R*, but could have any name – we have the JAGS (and in fact BUGS) code for this model. This is given by

```
 model {
    for (i in 1:N) {
          X[i]   ~ dnorm(theta, tau)
      }
theta ~ dnorm(1, 0.01*tau)
tau <- 1.0/sqrt(tau1)
tau1 ~ dgamma(1.0E-3, 1.0E-3)
 }

library(rjags)
X <- rnorm(50, mean=0, sd=100)
N <- 50
model2 <- jags.model(file="jagsmodel2.R", n.adapt=0)
update(model2, n.iter=5000)
output <- coda.samples(model2, c("theta","tau"), n.iter=10000)
plot(output)
```

which would gives the plot shown in Fig. 5.9

A few points to note from Example 5.4.2. Firstly you need to be careful to note that the way R and JAGS codes is subtly different in the choice of parameters to use. Compare above `dnorm(theta, tau)` in JAGS, which uses the mean and precision (inverse of variance), while R uses `rnorm(50, mean=0, sd=100)` which uses mean and standard deviation. Further the

[2]For instructions on downloading JAGS, see the home page at http://mcmc-jags.sourceforge.net.

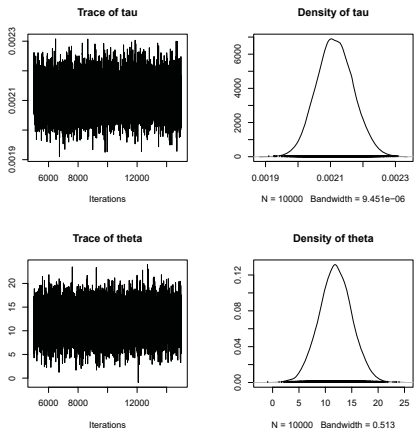

Figure 5.9: Output from Example 5.4.2

variable which the model uses (in the case of the example above `X,N`, need to lie in the R session.

Example (**5.4.1 revisited).** We can write the JAGS part of the code as

```
model {
#observation model
for (t in 1:N){
Y[t]~dnorm(mu[t],tau1);
}
#state model
for (t in 2:N){
mu[t] ~ dnorm(mu[t-1],tau2);
}
mu[1] ~dnorm(0, .0001); #initial value
tau2 ~dgamma(.001,.001); #precision
tau1 ~ dgamma(.001,.001) #precision
}
```

Figure 5.10 shows the posterior mean of μ_t in DLM model in red with the Nile data in black. The estimates for V and W come from the estimates for τ_1^{-1} and τ_1^{-2} in the DLM. We can also get the posterior distributions for V and W and these are shown in Fig. 5.10.

Suppose now we want to add covariates to the model. For example, we know that the Ashwan high dam was started in 1960 and it is possible that this is a change point in the time series.

Definition 5.4.3. (Simple dynamic regression and change points) In Definition 5.3.11 we saw how to at a covariate to a dlm. Suppose x_t is binary,

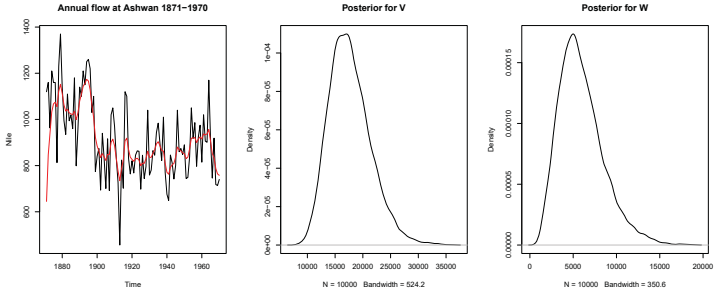

Figure 5.10: MCMC output from DLM for Nile data

with being zero before the change point and one after. We could build a model of the form

$$Y_t = \beta_{0,t} + \beta_1 x_t + \epsilon_t$$

where we allow θ to be slowing varying via

$$\beta_{0,t} = \beta_{0,t-1} + \omega_t.$$

where here we have let the intercept be dynamic but kept the change point parameter fixed.

Example (5.4.1 revisited). We can implement this in the Nile example. First lets look at the JAGS code:

```
model {
#observation model
for (t in 1:N){
Y[t]~dnorm(beta0[t]+beta1*X[t],tau1);
}
#state model
for (t in 2:N){
beta0[t] ~ dnorm(beta0[t-1],tau2);
}
beta0[1] ~dnorm(0, .0001); #initial value
beta1 ~dnorm(0, .0001); #change point effect
tau2 ~dgamma(.001,.001); #precision
tau1 ~ dgamma(.001,.001) #precision
}
```

This is run from inside R using

```
> library(rjags)
> Y <- Nile
> N <- length(Y)
> X <- as.numeric(time(Y) > 1898)
> model <- jags.model(file="jagsmodelchangepoint.R", n.adapt=0)
```

```
> update(model, n.iter=100)
> output <- coda.samples(model, c("beta0", "beta1", "tau1", "tau2"),
    n.iter=100000)
```

The output can be seen in Fig. 5.11 with the right hand plot showing the posterior for β_1. We can ask the question about if we think there is a change

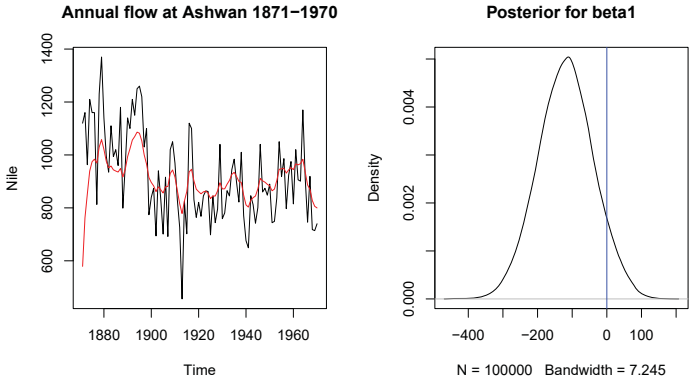

Figure 5.11: MCMC output from DLM for Nile data change point problem

point in terms of the posterior probability that $\beta_1 < 0$ i.e $P(\beta_1 < 0|data)$. In R use look at

```
> beta.list <- output[[1]][, 101]
> mean(beta.list < 0 )
[1] 0.92628
```

where we note that β_0 has 100 elements in this case, so β_1 is the 101 column in the output. Also we have used Approximation 5.5 where we approximate a probability (i.e. an integral) with a sample average. Also note that in this example the chain was run 100000 times to ensure 'convergence'. It is often the case with MCMC methods that the user has to play with the run time, `n.iter`. It is *not* a black box.

We can also look at the local trend model of Definition **??**. Here we have more structure to the dynamics.

Example (5.4.1 revisited). The JAGS code for this model is given by:

```
model {
#observation model
for (t in 1:N){
Y[t]~dnorm(mu[t],tau1);
}
#state model
for (t in 2:N){
mu[t] ~ dnorm(mu[t-1]+ beta[t-1],tau2);
```

```
beta[t] ~ dnorm(beta[t-1],tau3);
}
mu[1] ~dnorm(1120, .0001); #initial value
beta[1] ~dnorm(0, .0001); #initial value
tau3 ~dgamma(.001,.001); #precision
tau2 ~dgamma(.001,.001); #precision
tau1 ~ dgamma(.001,.001) #precision
}
```

Fig. 5.12 shows the fit. The right hand plot is the posterior mean of the local trend term. We note that the smallest value of the trend is exactly where the change point mentioned above was found, a linear term in the trend would correspond to a quadratic term in the level, also it is interesting to note that a second dam was build in the 1960 where again we see a change in the local trend. It may, though be unwise to over interpret this graph without further analysis.

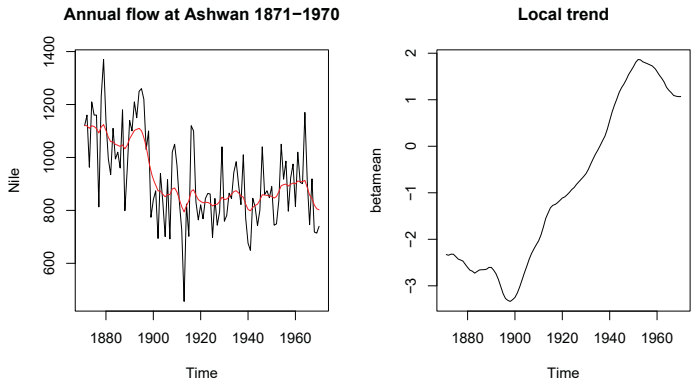

Figure 5.12: MCMC output from the local trend DLM for Nile data

5.5 State space representations of ARIMA models

The ARIMA models we studied in Chapter 4 can be written in a state space form and then the efficient prediction methods of the Kalman filter can be used directly. In fact the `predict.Arima( )`, `predict( )` and `arima( )` functions all use this method.

Example 5.5.1. (State space representation of $AR(1)$ process) We can write a AR(1) model

$$X_t = \phi X_{t-1} + Z_t$$

where $Z_t \sim N(0, \sigma^2)$ as a state space model trivially as

$$\begin{aligned} Y_t &= X_t, \\ X_t &= \phi X_{t-1} + Z_t. \end{aligned}$$

Example 5.5.2. (State space representation of $AR(2)$ process) We can write a AR(2) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

where $Z_t \sim N(0, \sigma^2)$ as a state space model by having a two dimensional state variable $\boldsymbol{X}_t = (X_t, X_{t-1})^T$ and

$$\begin{aligned} Y_t &= \begin{pmatrix} 1 & 0 \end{pmatrix} \boldsymbol{X}_t, \\ \boldsymbol{X}_t &= \begin{pmatrix} \phi_1 & \phi_2 \end{pmatrix} \boldsymbol{X}_{t-1} + Z_t \end{aligned}$$

Definition 5.5.3. (`makeARIMA( )` function) In R if you want the state space representation of an $ARIMA(p, d, q)$ model we can use `makeARIMA( )` function. This will output a state space model with the notation

$$\begin{aligned} y &= Za + \eta, \eta \sim N(0, h) \\ a &= Ta + Re, e \sim N(0, Q) \end{aligned}$$

it can then be used, for example in the `KalmanLike( )` function to compute a likelihood.

5.6 Case Study

5.6.1 Introduction

The TSUNAMI group of insurance and reinsurance companies have made available various datasets which cover issues important for the insurance industry. This report concentrates on a set of a seven year series of commercial claims due to weather. The TSUNAMI group in particular have an interest in the following questions.

1. *Temporal effects.* Is there any significant variation across the years in the observed data, apart from that caused by inflation? Any such variation should be identified and described.

2. *Aggregation.* There is interest in the aggregation in the data, usually over three day periods. Suitable models for such summary statistics need to be found.

3. *Extremes.* The data set contains two 'extreme' events. These were the storms in January and February 1990. The modelling should be able to predict return periods for extreme events such as these.

The report is structured in the following way. Section 5.6.2 contains details of both the currently available dataset and important data which is potentially available. Section 5.6.3 examines the modelling approach used throughout, pointing out the differences with traditional time series models which we regard as inappropriate for this kind of data. Section 5.6.4 looks at the results of some fitted models, while the final section describes future work.

5.6.2 Data

The data covers a period from 1st January 1990 until 31st December 1996. It is a record of all claims for UK commercial policies which are coded as being caused by 'weather'. For each claim the following information is available: the date of the reported claim incidence, the settlement date, and the claim amount. It is known that the data exhibits structure which is induced by the systematic way incidence dates are allocated when the exact date is unknown, forgotten or censored. For example, there are relatively more claims on the first day of each month, and there are relatively fewer claims at the weekend. It is important that the modelling takes into account this structure.

Totalling all the claims occurring on a particular day creates a time series of total daily claims. This series for both the raw data and the log of the data are shown in Figure 5.13. Alternatively the data can be represented in its original format, i.e. individual claim sizes and incidence dates, and an additional time series created which contains the number of claims made on each day. These are shown in Figures 5.14 and 5.15 respectively.

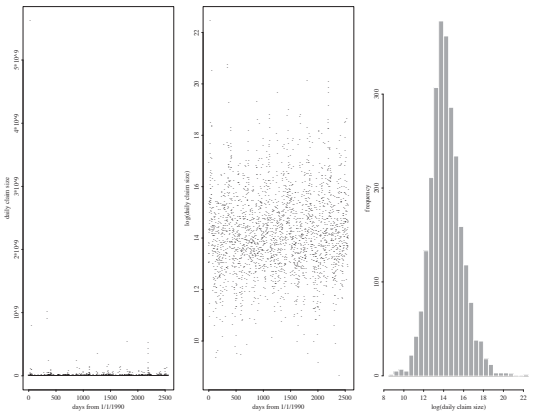

Figure 5.13: The total daily claim sizes from 1/1/1990 until 31/12/1996.

It is also important to consider information which, while unavailable at this stage of the modelling process, is in principle available to individual

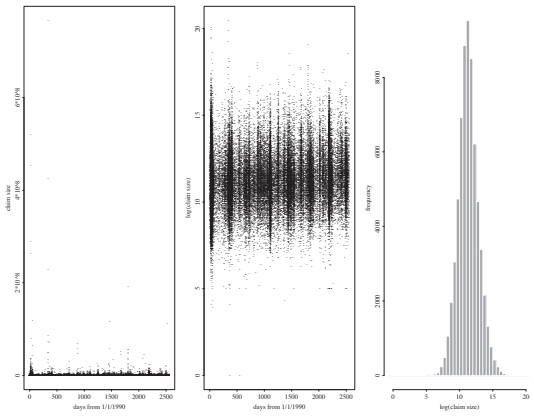

Figure 5.14: The individual claim sizes on each day from 1/1/1990 until 31/12/1996.

insurance companies and which will be clearly important in meeting the objectives in Section 5.6.1. This missing information will be important in the following issues.

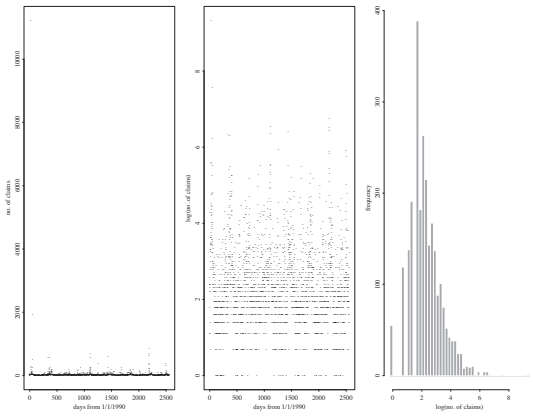

Figure 5.15: The number of claims on each day from 1/1/1990 until 31/12/1996.

1. *Temporal effects.* Currently not recorded are the total number of policies providing cover on each day. This series is important in understanding the change in behaviour over time of the data with which it is confounded.

2. *Aggregation.* Understanding the dependence structure across the dif-

ferent claims is an important part of the modelling. Information which would be useful here includes the geographical location of claims, the type of damage and the type of property concerned. This will be useful since when looking at aggregation it would be informative to know when different claims may be due to the same metrological event. For example flood and wind damgage may well have quite different aggregation properties.

3. *Extremes.* The geographical and claims information mentioned above will also be important in the modelling of extreme events. For example the relationship between the intensity of the metrological event and its location will contribute to both the number and size of observed claims.

The models developed below attempt to take account of this missing information by using random effects. Further they are designed to extend naturally when and if the above data becomes available.

5.6.3 The modelling approach

The underlying structure of the data, with random numbers of claims on each day, is unsuitable for traditional time series methods. Therefore in this paper we follow an *events based* approach. The physical cause of the damage claims will always be some metrological event, for example a storm or flood. Although these events are unobserved the structure of the model should take them into account. In particular a high proportion of the variation in the size of claims depends only on the weather on each day. We therefore model these metrological events by using random effects modelling.

In general we are interested in identifying the joint distribution of the claim sizes and the number of claims on any particular day. This allows us to address all the issues of interest raised in Section 5.6.1. We denote the claims sizes by X, the number of claims by N, the day of interest as t, and the joint distribution on this day as $f_t(X, N)$. We then modelled this joint distribution as a product of the conditional distribution of X given N and the marginal distribution of N, i.e.

$$f_t(X, N) = f_t(X|N)f_t(N).$$

Note that we will apply the events based approach by concentrating on the distribution of N. Informal data analysis shows that after conditioning on N the distribution of X remains approximately constant across time. This can be interpreted as meaning that the information in the metrological events is filtered primarily through the number of claims.

Models for $f_t(N)$

Let N_t denote the number of claims on day t. We shall consider the distribution of N_t conditionally on both the weather for that day and the number of policies at risk. Standard modelling assumptions, Feller (1966), which rely on the independence of claims given the weather conditions dictate that N_t will follow a Poisson distribution if we further assume that the actual proportion of policies which give rise to a claim on this day is small. Thus we apply the approximation of a Poisson distribution with a rate parameter λ_t. So we have

$$N_t \sim \text{Poisson}(\lambda_t).$$

Note that the information about the number of policies acts as a multiplier for the rate parameter λ_t.

Since the actual weather and number of policies are unobserved we will treat these factors as random effects for each day. A large proportion of the information which determines the parameter λ_t represents the effects of the weather on day t. Clearly this will have a component of seasonal variation. It also will be correlated in time since a weather system typically takes about three days to pass across the country. Further the systematic structure of reporting claims mentioned in Section 5.6.2 will affect the value of λ_t, for example it should be higher on the first day of each month. Putting these factors together means we assume a model

$$\log \lambda_t = H_t + \epsilon_t,$$

where H_t is a deterministic function of time which represents the fixed temporal effects, while ϵ_t represents the random weather effects. To model the correlation across time we shall assume a time series structure for ϵ_t. As an example a simple possibility is

$$\epsilon_t = \rho\epsilon_{t-1} + \nu_t \tag{5.6}$$

where $\nu_t \sim N(0, \sigma_t^2)$.

All models in this paper will be based on this structure. We therefore examine in Section 5.6.4 the appropriate form of the deterministic effect, H_t, and the time series structure of the random effects ϵ_t.

Models for $f_t(X|N)$

Data analysis indicates that after conditioning on the number of claims there is very little seasonal variation in the distribution of X. It appears that a relatively simple model, such as a fixed t-distribution is appropriate here.

5.6.4 Results

Throughout we use a Bayesian approach implemented using Markov chain Monte Carlo (MCMC) methodology, see Gilks, Richardson, and Spiegelhalter (Gilks et al.) and Gelman et al. (2014). As far as possible non-informative priors were used, and in general the likelihood completely dominated the posterior distribution which made the choice of prior relatively unimportant. There was one exception to this mentioned in Section 5.6.4. In general a sequential Metropolis-Hastings algorithm was used, although for some of the models we used block-updating to increase efficiency, see Gilks, Richardson, and Spiegelhalter (Gilks et al.). The fact that there was a large amount of data and that the models were clearly identified meant that the convergence of the MCMC algorithms gave few problems after parameterisations were chosen to aid efficiency, Brooks and Roberts (1998) and Cowles and Carlin (1996).

During the model selection process we attempted to keep the dimension of the model as small as possible, using the posterior distribution of both the parameter values and the likelihood as a guide.

The deterministic model

The deterministic part of the model H_t needs to include long term changes of the level, seasonal effects and the systematic features of the reporting mentioned in Section 5.6.2. This is currently modelled as

$$H_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 f^m(t) + \sum_{i=1}^{7}(\beta_i d_i(t)) + \sum_{i=1}^{12}(\gamma_i m_i(t)), \qquad (5.7)$$

where f^w is an indicator function which is 1 if t is the first day of the month, and 0 otherwise. While m_i is the indicator function for the i^{th} month of the year and d_i for the i^{th} day of the week.

We first discuss the long term trends. The output from the model is shown in Figure 5.16. The left hand panels show the output of the MCMC algorithm for each of the three parameters, α_0, α_1 and α_2. They show that the algorithm is mixing well and the output will give the posterior distribution of the parameters. The convergence seen here is typical of that for the other parameters in the model. The right hand panels show the marginal posterior distribution of the parameters. They show that there is reasonable evidence for both α_1 and α_2 being non zero. However it is important to note that the unobserved number of policies would enter as an additive part of equation (5.7), hence will be confounded with these trend terms.

The seasonal effects also were found to be strongly significant in all models. Various versions of a season component were tried including sinusoidal components. The twelve level monthly model used in equation (5.7) was found to be a simple and effective description of this variation. Figure 5.17

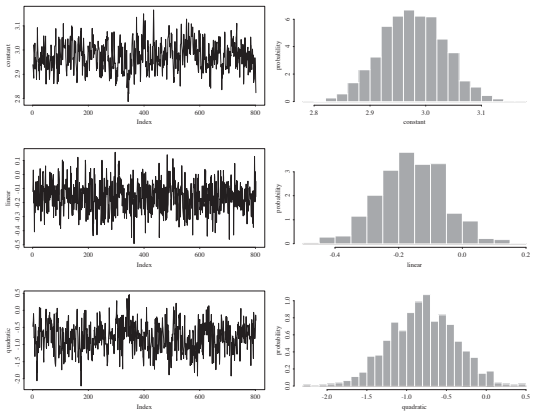

Figure 5.16: The trend effect: left hand panel shows output from MCMC algorithm, right hand panel showing marginal distribution.

shows the posterior output for the twelve levels. The seasonal effect, with winter being worse than summer, is very obvious.

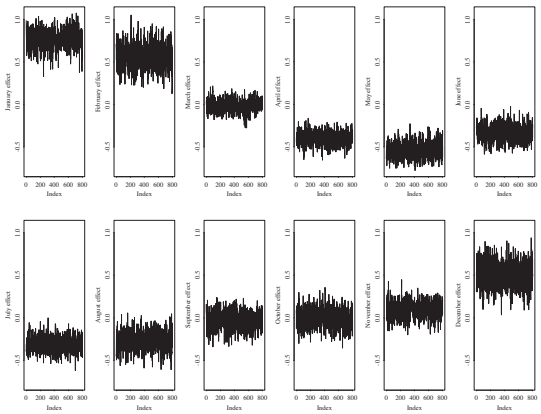

Figure 5.17: The seasonal monthly effect

The systematic reporting effects are picked up by the day of the week effect, shown in Figure 5.18, and the first day of the week effect, shown in Figure 5.19. For the weekly effect we see that there is an excess of claims on Monday and a corresponding decrease at the weekend. The first day of the week effect is shown to be significantly positive.

Overall the model has done very well in exploring the deterministic variation across days. It can detect long term variation and has quantified the systematic reporting effects very clearly.

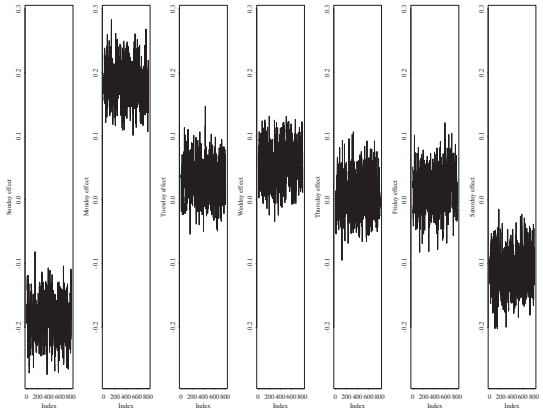

Figure 5.18: The weekly effect

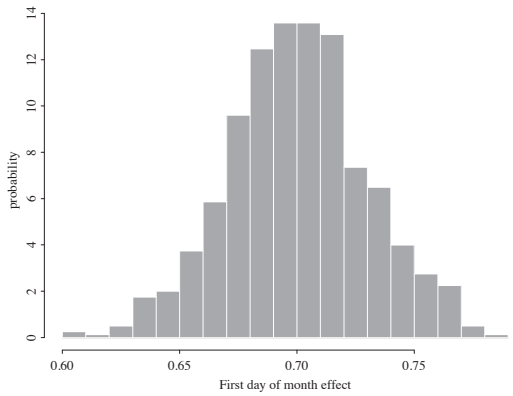

Figure 5.19: The first day of the month effect

The random effects

The modelling for the random effects needs to take into account both the marginal and the dependence structure. A simple example of a model is given by Equation (5.6) which simply models the correlation across time using a simple autoregressive structure. The innovation process ν is given by a Normal model. The variance of this term is modelled as having seasonal variation as

$$\sigma_t^2 = \sigma^2 \sum_{i=1}^{12} \gamma_i m_i(t)$$

where m_i is the monthly indicator function defined in Equation (5.7). The seasonal variation in the variance appears to be strongly significant.

In Figure 5.20 we plot the marginal distribution of the innovation process, based on this model. The left hand panel shows the marginal structure. The histogram shows a skew symmetric distribution. This is also shown in the plot of the innovations against the day in the right hand panel. The very large values in this plot correspond to very large jumps in one day in the number of claims.

The model above is capable of modelling the bulk of this distribution well, but there is a question of how to model both of the tails. Other innovation processes which have been used include a t-distribution, and mixtures of Normals and t-distributions. For further details of this issue see Section 5.6.5.

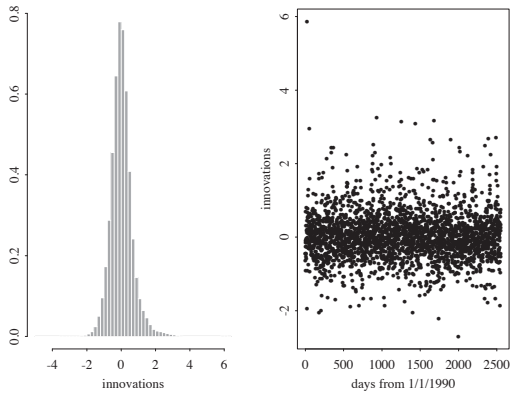

Figure 5.20: The innovation process; the left hand panel shows the marginal distribution, the right hand panel the time series structure

Model checking

One of the advantages of the MCMC methodology is that simple data analytic techniques can be used to give informative model diagnostics. One useful technique is to use the predictive distribution of the model and compare the output with the empirical distribution based on the data. As an example of this technique see Figure 5.21. This a comparison of the predictive distribution to the observed data. In this example we look at the proportion of days with less than 10 claims. The histogram is the predictive distribution generated from the model. The vertical line is the observed number of claims. We see that this is completely consistent with the models predictions.

5.6.5 Conclusions and further work

The modelling methodology here is designed to reflect the physical process which is generating the data. Traditional time series approaches are forced to model some aggregation of the data, by modelling directly the total cost of claims on a day, for example. In contrast we model the full joint distribution of all the data. We feel that a model which is closer to the actual physical process offers great advantages in interpretation and predictive power.

There are a number of outstanding issues which need to be explored. We look at the three objectives raised in Section 5.6.1.

1. *Temporal effects.* The model is able to detect yearly variation, both using trends in the deterministic function, H_t, and potentially in the structure of the innovation process. The model recognises that changes in the numbers of claims is confounded with changes in the number of policies at risk. However there is scope to incorporate this statistic in order to give a satisfactory model for this first question.

2. *Aggregation.* By modelling the full joint distribution we have implicitly modelled any aggregation process. This can be done either analytically or via simulation. Clearly there is further work to be done here.

3. *Extremes.* The most delicate part of the modelling concerns the incorporation of the extreme metrological events in the dataset. Currently different models of the innovation process can treat this observations very differently. Further work is clearly needed in order to clarify this issue. A more detailed dataset, as mentioned in Section 5.6.2 could help. Since we have modelled the data via the physical process driving it, there is also the possibility of using metrological models directly in the modelling. This possibility opens up a whole new set of information which can greatly improve prediction of return levels of these extreme events.

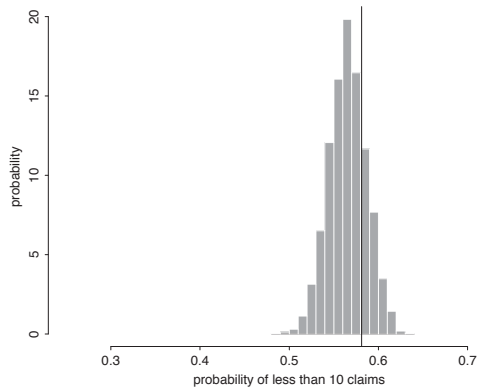

Figure 5.21: An example of comparing the predictive distribution to the observed data. In this example we look at the proportion of days with less than 10 claims. The vertical line is the observed number of claims

6

Other topics in time series modelling

6.1 Introduction

In this chapter we take a *brief* look at a number of other topics that are useful in time series modelling. We will not be looking at these in great depth, rather give pointers to where you can find ideas if you are interested.

6.2 The Kalman filter

6.2.1 Historial note

Having looked at dlm, and their Bayesian analyses, we now look at a related idea, the Kalman filter. While these are mathematically strongly related they do, in fact have a very different background. The filter, taking its name from the paper Kalman (1960), had it origins in engineering control problems and found early applications in controlling spacecraft in the Apollo program. The relationship between forecasting and control is a strong one – indeed the point of many forecasting exercises is to ask the 'what if?' question to see if a system needs to be changed in some way. These types of questions about engineering control also motivated many of the ideas of Box and Jenkins and their book, Box and Jenkins (1976), is titled *Time series analysis: forecasting and control*, showing the importance of the link in their opinion.

In many of these applications the speed of computation was one of the most important features. With real time control problems you typically need to be able to compute predictions faster than you are gathering data. Think about controlling a moon-landing for example. This concern with speed is reflected in the use of many recursive computation methods. Although computational power has increased exponentially since the development of these

ideas there are still many current applications where speed of computation is critical. The so-called high-frequency trading in finance is one such case.

6.2.2 Kalman filter

The link between Bayesian methods and the Kalman filter can be found in Appendix 6.5.3. Despite these results we shall look at the Kalman filter from first principles.

One problem with the Kalman filter is that there is not agreement on standard notation. We will therefore define models again for clarity. The review paper Sorenson (1970) looks at the relationship between Gauss's (and Legendre's) least squares method and the Kalman filter. It gives a nice history of the Kalman filter linking it to the work problem solved by Kolmogorov and Weiner – although this solution involved spectral (Laplace and Fourier methods), rather than state space, methods.

Definition 6.2.1. (State space model)
If we use the notation of Sorenson (1970) we have the state space model

$$\text{Observation}: \quad z_k = H_k x_k + v_k \qquad v_i \overset{i.i.d.}{\sim} N(0, R_k) \tag{6.1}$$

$$\text{State}: \quad x_{k+1} = \Phi_{k+1,k} x_k + w_k, \qquad w_k \overset{i.i.d.}{\sim} N(0, Q_k) \tag{6.2}$$

and where the noise terms v and w are independent of each other. We have initial conditions of the form $x_0 \sim N(\widehat{x}_{0|-1}, P_{0|-1})$.

In this definition we assume that the terms $H_k, \Phi_{k+1,k}, R_k$ and Q_k are all known. This is different from the dynamic linear model set up where these are estimated. The following example shows a case where this is reasonable.

Example 6.2.2. (Physics example) Suppose we have a body falling in a vacuum and its height, at time t –which to start with is continuous – is $y(t)$. We have, from physics theory, that

$$\frac{d^2y}{dt^2}(t) = -g,$$

and we can solve this differential equation to give

$$y(t) = y(t_0) + \frac{dy}{dt}(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2.$$

where t_0 is some initial time. If we now discretise time so that we have $t - t_0 = 1$ we have, slightly changing notation, by setting $y(t) = y_t$ and $\frac{dy}{dt} = \dot{y}_t$,

$$y_{t+1} = y_t + \dot{y}_t - \frac{g}{2}.$$

Let us therefore define the state vector $x_t := (y_t, \dot{y}_t)^T$ and so, physics gives us the state equation

$$x_{k+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x_k - \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} g$$

which are the equations

$$\begin{aligned} y_{t+1} &= y_t + \dot{y}_t - \frac{g}{2}, \\ \dot{y}_{t+1} &= \dot{y}_t - g. \end{aligned}$$

Now, suppose we have some experimental measuring device which records a (noisy) measurement of x_k. Then we have the observation equations

$$z_t = H_k x_k + v_k.$$

The terms H_k and the distribution of v_k would be found by calibration experiments, and may –or may not – agree with the assumptions of Definition 6.2.1.

In this example the laws of physics and calibration experiments give us terms that in the DLM section we where using Bayesian methods to estimate just from the data. These laws and experiments are clearly highly informative and information here should be used for forecasting and filtering.

Definition 6.2.3. (Kalman filter) Using the terms in Definition 6.2.1 we can think about the filtering problem. Let us define $\widehat{x}_{k|k}$ to be the best mean square estimate of x_k based on $z_0, z_1, \ldots, z_k$ while $\widehat{x}_{k|k-1}$ is the best estimate based on $z_0, z_1, \ldots, z_{k-1}$.

We work recursively first compute

$$\widehat{x}_{k|k-1} = \Phi_{k,k-1} \widehat{x}_{k-1|k-1} \tag{6.3}$$

which holds since this is a linear equation based on the same information $z_0, z_1, \ldots, z_{k-1}$. and the uncertainty in this estimate is measured by

$$P_{k|k-1} := Var(x_k - \widehat{x}_{k|k-1}) = \Phi_{k|k-1} P_{k-1|k-1} \Phi_{k|k-1}^T + Q_{k-1}. \tag{6.4}$$

The Equations (6.3) and (6.4) are called the *prediction* part of the Filter.

We now define the *residual* $r_k := z_k - H_k \widehat{x}_{k|k-1}$ and use it to update the estimate to be

$$\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k \left[z_k - H_k \widehat{x}_{k|k-1} \right] \tag{6.5}$$

The weighting term K_k is called the *Kalman gain*. This is found by minimising the sum of squares and it can be show that

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k \right)^{-1} \tag{6.6}$$

where

$$P_{k|k} := Var(x_k - \widehat{x}_{k|k}) = (I - K_k H_k)\, P_{k|k-1} \tag{6.7}$$

The last set of equations – (6.5) (6.6) and (6.7) – are called the *correction equations*. These use the prediction error to correct the original forecast.

In R functions to run the filter can be found in the package `FKF` (fast Kalman filter) and the functions `fkf( )`. The notation is a little different so, for clarity we state the notation again

Definition 6.2.4. (General state space model in R) Let the (unobserved) state vector be denoted by $\alpha = \{\alpha_t\}$ and the observed process $y = \{y_t\}$.

$$\begin{aligned} y_t &= c_t + Z_t\alpha_t + G_t\epsilon_t \\ \alpha_t &= d_t + T_t\alpha_{t-1} + H\eta_t \end{aligned}$$

where $\eta \sim N(0, h)$ and $\epsilon_t \sim N(0, \kappa h)$ are independent and Z, T, R, h are given and κ controls the balance between the two variances.

We can think of this model for $\{y_t\}$ being parameterised by

$$(\mu_0, P_0, c_t, Z_t, G_t, d_t, T_t H_t)$$

where α_0 is the initial value with its uncertainty given by the variance matrix P_0. The model for the complete data $\{y_t, \alpha_t\}$ will clearly be high dimensional multivariate normal – for properties of multivariate normal see Appendix 6.5.2. We only see the y_t variable so we need to marginalise – integrate out – the α_t variable. This will still leave a high dimensional multivariate normal distribution for y_t. In principle we could write down the likelihood directly for this. We do not do this because it would involve inverting a large variance-covariance matrix and each time we get a new observation a new, and larger, matrix needs to be inverted. Instead we use recursive methods.

Example 6.2.5. (The Kalman filter in R) Here is an example of running the `fkf( )` function with the model

$$\begin{aligned} y_t &= 0 + 3\alpha_t + \sqrt{1}\epsilon_t, \\ \alpha_t &= 0 + 1\alpha_{t-1} + \sqrt{0.1}\eta_t \end{aligned}$$

The explicit recursions for the Kalman filter are given in Appendix 6.5.2 and you set up the model using the following code.

```
> y <- as.vector(Nile)
> n <- length(y) #number observations
> m <- 1 #dim state variable
> d <- 1 #dim observations
> a0 <-c(1) #initial  observations
> P0 <- matrix(c(4), nrow=m,ncol=m) #initial uncertainty
```

```
> dt <- matrix(c(0), nrow=m,ncol=n) #intercept state
> Tt <- array(c(1), dim=c(m,m,n)) #AR term state
> HHt <- array(c(0.1), dim=c(m,m,n)) #Varaince eta
> ct <- matrix(0, nrow=d, ncol=m) #intercept observation
> Zt <- array(c(3), dim=c(d,m,n)) #Regression
> GGt <- array(c(1), dim=c(d,d,n)) #Varinace epsilon
> yt <- matrix( y, nrow=d, ncol=n) #Data
> out <- fkf(a0, P0, dt, ct, Tt, Zt, HHt, GGt, yt, check.input = TRUE)

> par(mfcol=c(1,2))
> plot(y, main="fkf output")
> lines(0 + 3*out$att[1,], col="red")
> plot(out$vt[1,], main="prediction errors", ylab="errors")
> out$logLik
[1] -455510.3
```

Figure 6.1 shows the output. More details of the function can be found in Appendix 6.5.2.

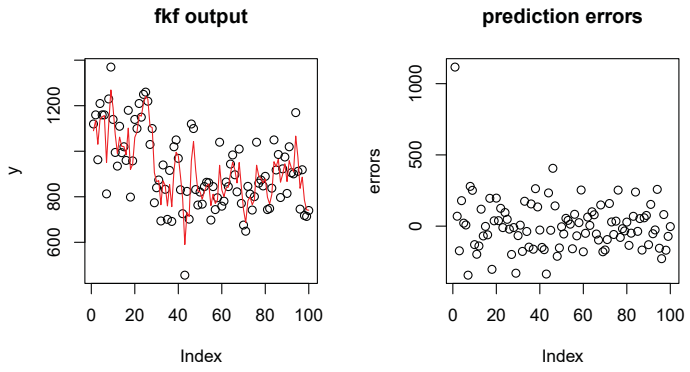

Figure 6.1: Some output from `fkf()`

6.3 ARCH and GARCH modelling

Example 6.3.1. (The Bollerslev-Ghysel benchmark dataset)

The data in Fig. 6.2 from Bollerslev and Ghysels (1996) is the daily percentage nominal returns for the Deutschemark-Pound exchange rate. We see the non-constant volatility and volatility clustering that is common in such financial data. We also can see from the marginal plots that, where the dynamics is excluded, we get non-normal behaviour. The data is 'heavier-tail' than a constant variance Gaussian process.

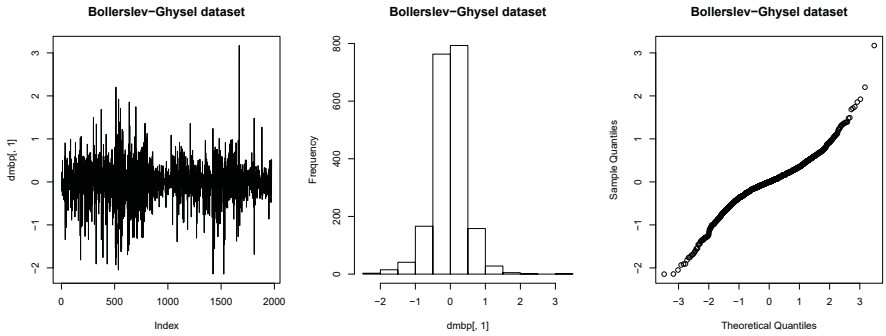

Figure 6.2: Bollerslev-Ghysel benchmark dataset

We have already seen, in Chapter 1, a way of modelling data which has non-constant volatility. We recall the definition here.

Definition 6.3.2. (ARCH model) An AutoRegressive Conditional Heteroscedasticity (ARCH(p)) model is defined hierarchically: first define $X_t = \sigma_t Z_t$ where $Z_t \sim N(0,1)$ i.i.d., but treat σ as being random such that

$$\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \cdots + \alpha_p X_{t-p}^2$$

So the variance is time dependent – a large value of X_t will result in period of high volatility.

We can write X_t as a *non-linear* function of the previous innovations in the following way. Note that this is different from the ARMA story where, if we have a causal stationary process we write it as a *linear* function of the previous innovations. Let just look at the ARCH(1) model. We have, following (Brockwell and Davis, 2002, Page 350),

$$X_t = \sigma_t Z_t, \sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2.$$

So we have

$$\begin{aligned} X_t^2 &= \sigma_t^2 Z_t^2 = \alpha_0 Z_t^2 + \alpha_1 Z_t^2 X_{t-1}^2 \\ &= \alpha_0 Z_t^2 + \alpha_1 Z_t^2 \left(\alpha_0 Z_{t-1}^2 + \alpha_1 Z_{t-1}^2 X_{t-2}^2\right) \\ &= \alpha_0 Z_t^2 + \alpha_0 \alpha_1 Z_t^2 Z_{t-1}^2 + \alpha_1^2 Z_t^2 Z_{t-1}^2 X_{t-2}^2 \\ &= \alpha_0 Z_t^2 + \alpha_0 \alpha_1 Z_t^2 Z_{t-1}^2 + \alpha_0 \alpha_1^2 Z_t^2 Z_{t-1}^2 Z_{t-2}^2 + \cdots + \alpha_1^{n+1} Z_t^2 Z_{t-1}^2 \cdots Z_{t-n}^2 X_{t-n-1}^2 \end{aligned}$$

Now if we assume $\alpha_1| < 1$ this last term will converge to zero as $n \to \infty$ and we get the expression

$$X_t^2 = \alpha_0 \left(Z_t^2 + \alpha_1 Z_t^2 Z_{t-1}^2 + \alpha_1^2 Z_t^2 Z_{t-1}^2 Z_{t-2}^2 + \ldots\right).$$

This is the non-linear causal representation of X_t in terms of previous Z_s innovation. Taking expectation gives

$$E\left(X_t^2\right) = \alpha_0\left(1 + \alpha_1 + \alpha_1^2 + \dots\right) = \frac{\alpha_0}{1-\alpha_1}.$$

As with ARMA models we need to check if there are stationary solutions for any given sets of parameters. Using these results gives the following conclusion.

Theorem 6.3.3. (Existence of stationary solution) For an ARCH(1) model we have, when $|\alpha_1| < 1$ there is a unique causal stationary solution of the equations of (6.3.2). It has the moment structure:

$$\begin{aligned} E(X_t) &= E\left(E(X_t|Z_s, s<t)\right) = 0, \\ Var(X_t) &= \frac{\alpha_0}{1-\alpha_1}, \\ Cov(X_{t+h}, X_t) &= E\left(E(X_{t+h}X_t|Z_s, s<t+h)\right) = 0, \end{aligned}$$

for $h > 0$.

Proof. Use the properties of iterated expectations. □

One consequence of Theorem 6.3.3 is that the terms X_i and X_j, for $i \neq j$, are uncorrelated, but they are dependent. It is only for the normal distribution that zero correlation implies independence and, as we shall see, ARCH models are not (marginally) normally distributed, rather they have heavier tails.

As might be expected we can generalise the ARCH model.

Definition 6.3.4. (GARCH model) The GARCH(p, q) model is defined by $X_t = \sigma_t Z_t$ where $Z_t \sim N(0,1)$ i.i.d. and

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2.$$

On important feature of GARCH modelling is if volatility is high for one time period, it will tend to stay high for a period of time. This is sometimes called 'persistence of volatility' or 'volatility clustering'.

There are, in fact, a very, very large number of variations on the theme on ARCH and GARCH, one place to look for details is Bollerslev (2008) and references therein, alternatively the book Francq and Zakoian (2011).

6.3.1 Working with GARCH models in R

For simulating ARCH or GARCH models it is straightforward to work directly in R. The following is just basic code which simulates an ARCH(1) model, and can be easily adapted for other orders or ARCH.

```
 arch.sim <- function(n, omega, alpha1, sigma)
{
out <- sqrt(omega)
for(i in 2:n)
{
out[i] <- sqrt(omega + alpha1*out[i-1]^2)*rnorm(1, sd=sigma)

}
out
}
```

We can see the output from this function for the output.

```
> sim <- arch.sim(2000, 10, 0.85, 1)
```

in Fig. 6.3. We can see, as expected from Theorem 6.3.3 that there is no significant structure in the `acf`-plot and also that the QQplot shows that the marginal distribution is indeed non-Gaussian, despite the fact that the innovations are generated from a normal distribution.

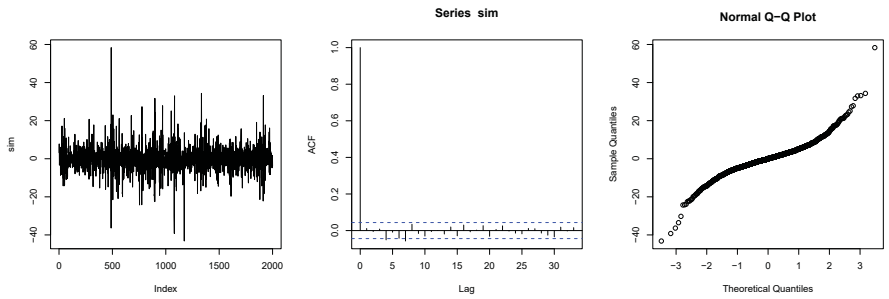

Figure 6.3: Simulated ARCH(1) data

While simulation can be very easily, and 'by hand', it is better to use specialist packages for fitting. In R we can use the `rugarch` package. This has a number of dependencies for example `DistributionUtils, GeneralizedHyperbolic, SkewHyperbolic,nloptr, misc3d, spd, truncnorm, Rsolnp, ks`, etc. You can either install libraries as R asks for them, or use the `install Dependencies` method when you install `rugarch`.

In the following examples we work with, what the package calls the standard GARCH model, `sgarch`, which is in fact slightly more general than Example 6.3.4 since the innovation term Z_t is allowed to be an ARIMA distributed random variable, and it does not have to have normal errors. In the following we fit to the simulated data

```
> library(rugarch)
```

```
>  Model2 <-  ugarchspec(variance.model = list(model = "sGARCH",
             garchOrder = c(1, 0) ),mean.model = list(armaOrder=c(0,0,0),
              include.mean = FALSE) )
> fit <-  ugarchfit(data = sim, spec = Model2)
> fit
 ---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,0)
Mean Model : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
omega   10.88071    0.596597   18.238        0
alpha1   0.83808    0.053021   15.806        0
```

As can be seen from the specification of `Model2` the `ugarchfit` library can fit much more complex GARCH models than we have defined. For example the mean model does not have to be an i.i.d. sequence, but can have its own time series structure which is defined in by the `mean.model` argument. Above it has been set to be the simplest possible. We such possible complexity in practice the model selection part of using GARCH models can be rather complex.

6.4 Frequency domain methods

6.4.1 Why the complex plane?

In this section we are going to be looking at the periodic part of time series analysis. We, of course, have looked at seasonality in previous sections but here we are going to be looking at the case where the important periods, or equivalently the frequency, of the time series are not known *a priori*. Because periodic functions, of different frequencies, are going to be key it is natural that we are going to be making use of trigonometric functions such as sines and cosines. What is perhaps less obvious is that we are also going to be using complex numbers. The following example motivates why it is very convenient to work in the complex plane. Recall that we link the

trigonometric functions and the complex plane through the identity

$$\cos(\theta) + i\sin(\theta) = e^{i\theta}. \tag{6.8}$$

To show how convenient it is to work in the complex plane consider the following proof of this important trigonometrical identity.

Theorem 6.4.1. We have the identity

$$\cos(u+v) = \cos(u)\cos(v) - \sin(u)\sin(v). \tag{6.9}$$

Proof. We write the left hand side, using Equation (6.8) as

$$\begin{aligned} \cos(u+v) &= \Re\left(e^{i(u+v)}\right) = \Re\left(e^{iu}e^{iv}\right) \\ &= \Re\left((\cos(u)+i\sin(u))(\cos(v)+i\sin(v))\right) \\ &= \cos(u)\cos(v) - \sin(u)\sin(v) \end{aligned}$$

□

This is representative of the ease of handling trigonometrical functions after switching to the complex domain. The identity (6.9) is also important since it shows how to change the phase of a trigonometrical function and one way of explaining why we use the complex domain is that it allows us to separate the frequency from the phase.

6.4.2 Spectral density and discrete Fourier transforms

We have seen, in Example 3.2.7 an example of a 'random' sine wave as a 'predictable process' i.e. where Z_1, Z_2 are two independent $N(0,\sigma^2)$ random variables and we define the discrete time series

$$X_t = Z_1\cos(2\pi t/100) + Z_2\sin(2\pi t/100),$$

which can be though of as a sine wave with random amplitude and phase by using the trigonometric identities such as

$$\sin(u+v) = \sin(u)\cos(v) - \cos(u)\sin(v), \sin(u) = \frac{Z_1}{\sqrt{Z_1^2+Z_2^2}}.$$

By allowing the frequency to vary we can have a class of time series

Example 6.4.2. (Linear combinations of sinusoids) Consider the process defined by

$$X_t := \sum_{i=1}^{k}\left(A_i\cos(2\pi\omega_i t) + B_i\sin(2\pi\omega_i t)\right) \tag{6.10}$$

where A_i, B_i are independent $WN(0,\sigma_i^2)$ processes.

Theorem 6.4.3. The auto-covariance function of (6.10) is

$$\gamma(h) = \sum_{i=1}^{k} \sigma_j^2 \cos(2\pi\omega_i h)$$

So the series is stationary.

Proof. We use the identity

$$\cos(u - v) = \cos(u)\cos(v) + \sin(u)\sin(v).$$

which can be proved easy in the complex plane. □

In order to understand the effect of the frequency ω_i on a discretely observed sinusoidal function consider Fig. 6.4. In these plots we have look at 50 discrete time points one unit apart. In the first plot the frequency is $\omega = 1/50$, so that across the observed points we see exactly one complete period of the sinusoidal function. This is called a low frequency element. Indeed for any lower frequency, and these time points, we could not observe a complete cycle. In the middle two panels we increase the frequency to 7/10 and 18/50 where we see that we now observe more cycles, but have less observations per cycle. The most extreme version of this is the bottom panel where the frequency is now 'high' at 1/2. At this level we only have two observations per cycle – the maximum and minimum.

We can therefore think of time series of the form (6.10) in terms of their high, medium and low frequency components.

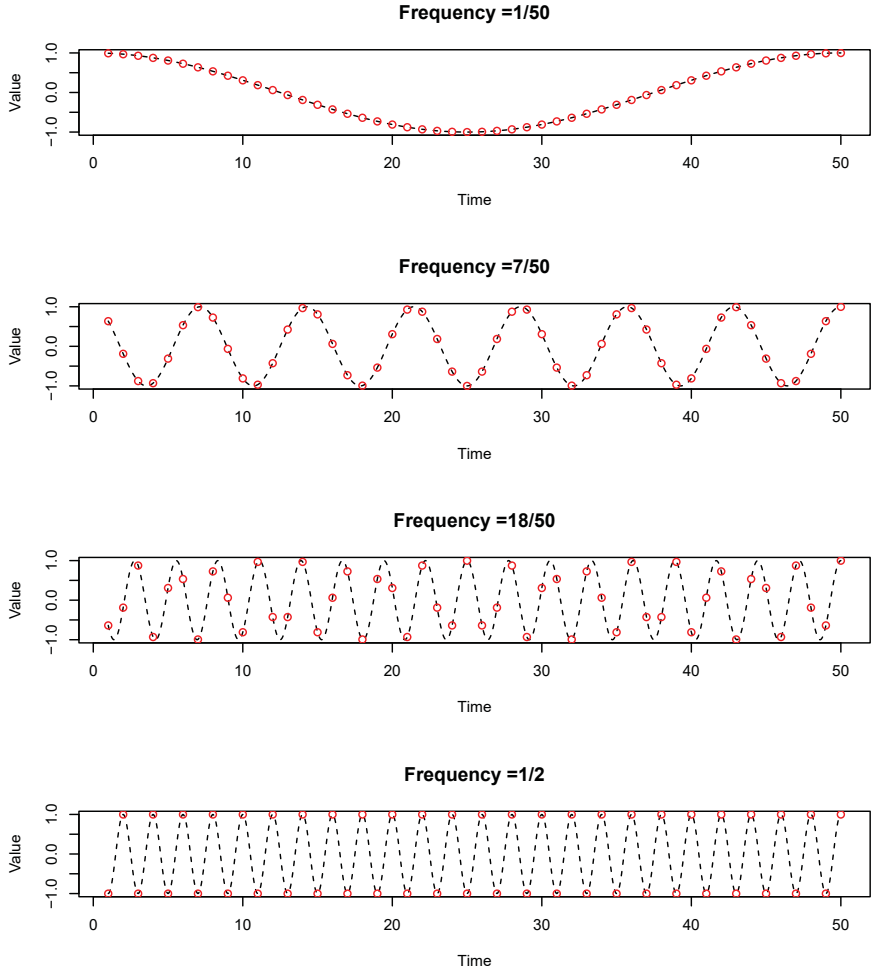

Figure 6.4: Discrete time series of simple periodic functions and the effect of changing the frequency

Example 6.4.4. (Linear combinations of sinusoids) Figure 6.5 shows two example of processes of the form (6.10) both with $k = 50$ and having 500 observed time points. They both have $\sigma_i^2 = 1$ for all i. They differ in the way the 50 values of ω_i where selected. Following the discussion above we have the constraints that $1/500 \leq \omega_i < 0.5$. In the top row the middle histogram summarises the values of ω_i. They are all 'low frequency' components and we can see that the time series has is varying slowly over the time steps. In contrast the lower row has selected the values of ω_i at the top of the range. These are high frequency and we see the time series has much more rapid oscillations over much shorter time scales.

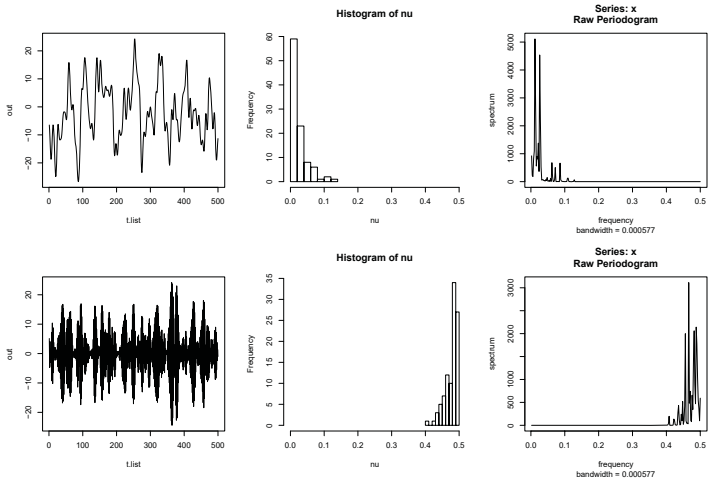

Figure 6.5: Simple periodic functions: top low frequency, bottom high frequency

The right most plots show the data based estimates of the distribution of the frequencies – the so-called spectral densities.

Definition 6.4.5. (Spectral density) If a mean zero, stationary time series $\{X_t\}$ has an auto-covariance function $\gamma(h)$ such that $\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$ then the *spectral density* is defined as

$$f(\nu) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \nu h}.$$

for $-\infty < \nu < \infty$.

Theorem 6.4.6. (Properties of spectral density) If $f(\nu)$ is the spectral density of a stationary time series it has the following properties

(i) For all ν

$$|f(\nu)| < \infty.$$

(ii) $f(\nu)$ is periodic with period 1, so we can define its domain as $-0.5 \leq \nu \leq 0.5$.

(iii) $f(\nu) = f(-\nu)$ is an even function. So we can effectively define the domain as $0 \leq \nu \leq 0.5$.

(iv) $f(\nu) > 0$

(v) Given the spectral density $f(\nu)$ we can reconstruct the auto-covariance function by the formula

$$\gamma(h) = \int_{-0.5}^{0.5} e^{2\pi i\nu h} f(\nu) d\nu \tag{6.11}$$

Proof. (Sketch) (i) The first property comes from the fact that $|e^{i\theta}| = 1$ and by using the absolute convergence of γ from Def. 6.4.5.

(ii) This follows since

$$e^{-2\pi\nu h} = e^{-2\pi\nu(h+1)}.$$

(iii) Follow from elementary properties.

(iv) First define

$$F_N(\lambda) = \frac{1}{N} E\left(\left| \sum_{r=1}^{N} X_r e^{-2\pi i r\lambda} \right|^2 \right) \geq 0$$

This can be expanded as

$$\begin{aligned} \frac{1}{N} E\left(\left| \sum_{r=1}^{N} X_r e^{-2\pi i r\lambda} \right|^2 \right) &= \frac{1}{N} E\left(\sum_{r=1}^{N} X_r e^{-2\pi i r\lambda} \sum_{t=1}^{N} X_t e^{2\pi i t\lambda} \right) \\ &= \frac{1}{N} \sum_{|h|<N} (N - |h|)\, e^{-2\pi i h\lambda} \gamma(h) \\ &\to \sum_{h=-\infty}^{\infty} e^{-2\pi i h\lambda} \gamma(h) = f(\lambda) \end{aligned}$$

as $N \to \infty$ So the spectral density is the limit of positive functions so is positive.

(v) If we swap the sum and integral in the expression we get

$$\begin{aligned} \int_{-0.5}^{0.5} e^{2\pi i\nu h} \left[\sum_{j=-\infty}^{\infty} \gamma(j) e^{-2\pi\nu j} \right] d\nu &= \sum_{j=-\infty}^{\infty} \gamma(j) \int_{-0.5}^{0.5} e^{2\pi(j-h)\nu} d\nu \\ &= \gamma(h). \end{aligned}$$

□

From Theorem 6.4.6 we see that restricting attention to the range $[0, 0.5]$ as discussed in Fig. 6.4 has a more formal justification.

Example 6.4.7. (White noise) The auto-covariance function for white noise, $WN(0, \sigma^2)$ is $\gamma(0) = \sigma^2$ and $\gamma(h) = 0$ for $h \neq 0$. Hence its spectral density is $f(\nu) = \sigma^2$, a constant.

In fact the term 'white noise' comes from this fact since this means all frequencies are equally represented in the spectrum as is true with 'white light'.

Example 6.4.8. (Spectral density for AR(1) process) For a stationary, causal AR(1) processes the spectral density given by

$$\begin{aligned} f(\nu) &= \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \nu h} \\ &= \frac{\sigma^2}{1-\phi_1^2} \sum_{h=-\infty}^{\infty} \phi_1^{|h|} e^{-2\pi i \nu h} \\ &= \frac{\sigma^2}{1-\phi_1^2} \left(1 + \sum_{h=1}^{\infty} \phi_1^h (e^{-2\pi i \nu h} + e^{2\pi i \nu h}) \right) \\ &= \frac{\sigma^2}{1-\phi_1^2} \left(1 + \frac{\phi_1 e^{-2\pi i \nu}}{1-\phi_1 e^{-2\pi i \nu}} + \frac{\phi_1 e^{2\pi i \nu}}{1-\phi_1 e^{2\pi i \nu}} \right) \\ &= \frac{\sigma^2}{1 - 2\phi_1 \cos(2\pi\nu) + \phi_1^2} \end{aligned}$$

We can see this in Fig. 6.6. The top row shows the case where $\phi_1 = 0.8$,with a time series on the left and the theoretical spectral density on the right shown with a dash line. The solid line in that panel is the estimated (and slightly smooth) spectral density. This is called the periodogram and is discussed below. We see that in this case there are a lot of low frequency components in the time series. We can contrast this with the middle row, which is $\phi_1 = 0$ or white noise. Now we have a flat spectrum with all frequencies represented. The bottom row is the case where $\phi_1 = -0.8$ and we expect many more high frequency components.

Example 6.4.9. (Spectral density for MA(1) process) If we have a MA(1) process then the spectral density is given by

$$f(\nu) = \sigma^2 \left(1 + \theta_1^2 + 2\theta_1 \cos(2\pi\nu) \right).$$

6.4.3 The Periodogram

If we just have data we might want to estimate the spectral density directly, rather than by using modelling assumptions. One way to do that is to use

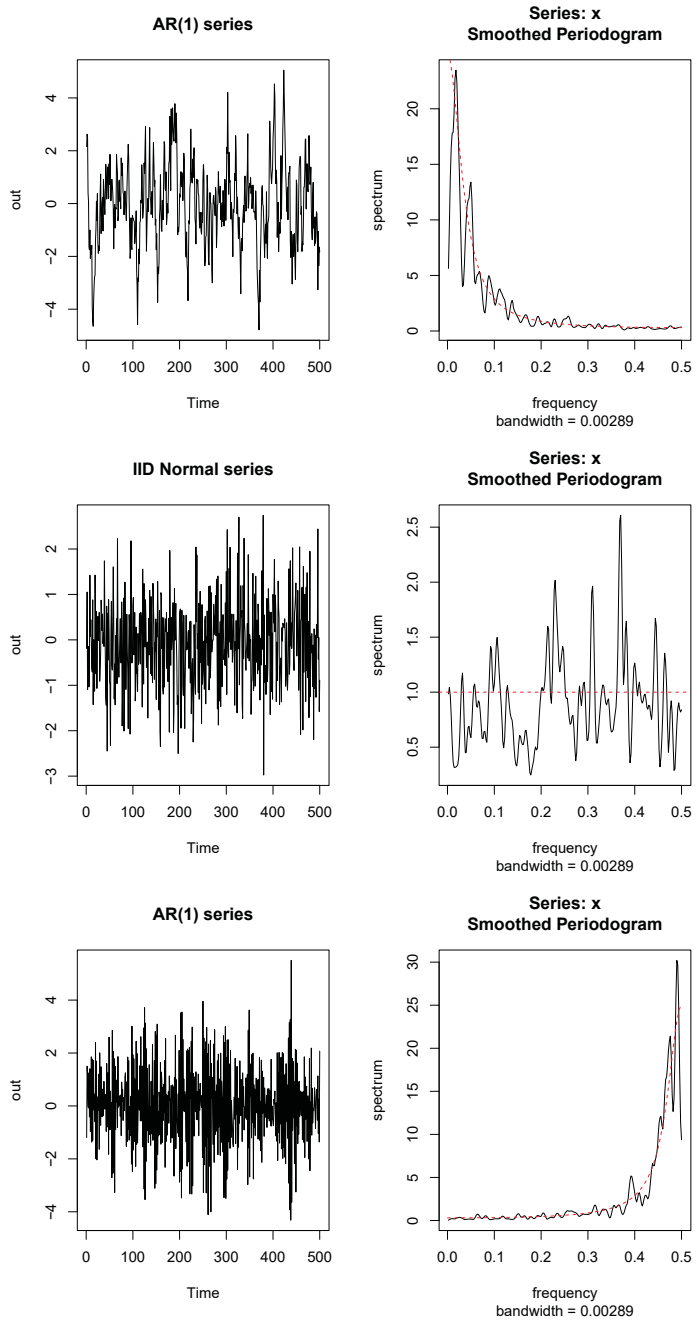

Figure 6.6: AR(1), $\phi = 0.8, 0, -0.8$ model, time series, smooth periodogram and spectral density

the periodogram or, in practice the smoothed periodogram. Before we define this we need some linear algebra.

Definition 6.4.10. (Complex inner product space) Consider the (complex) vector space $\mathbb{C}^n$ and the inner product $\langle z_1, z_2\rangle := z_1^* z_2$ where z^* is the transpose of the vector of complex conjugates of $z \in \mathbb{C}^n$. The corresponding norm is then

$$\|z\| := \langle z, z\rangle = \sum_{i=1}^{n} z_i \bar{z}_i = \sum_{i=1}^{n} |z|^2.$$

While the set of possible frequencies that we might want to study is a continuous one since we only have a finite set of data, of size n, we must restrict attention to smaller sets. One very useful choice is to look at the following finite set.

Definition 6.4.11. (Fourier frequencies) The Fourier frequencies are defined by $\nu_k = k/n$ for $k = 0, \ldots, n-1$.

Definition 6.4.12. We can define an orthonormal basis for $\mathbb{C}^n$ with the inner product defined in Def. 6.4.10 via

$$e_j = \frac{1}{\sqrt{n}} \left(e^{2\pi i \nu_j}, e^{2\pi i 2\nu_j}, \ldots, e^{2\pi n \nu_j} \right)^T$$

for $j = 0, \ldots, n-1$. We note that

$$\begin{aligned}
\langle e_j, e_k\rangle &= \frac{1}{n} \sum_{t=1}^{n} e^{-2\pi i t \nu_j} e^{2\pi i t \nu_k} \\
&= \frac{1}{n} \sum_{t=1}^{n} e^{-2\pi i t (\nu_k - \nu_j)} \\
&= \frac{1}{n} \sum_{t=1}^{n} e^{-2\pi i t (k-j)/n} \\
&= \frac{1}{n} \sum_{t=1}^{n} \left(e^{-2\pi i (k-j)/n} \right)^t \\
&= \begin{cases} 1 & k = j \\ \frac{1}{n} e^{-2\pi i (k-j)/n} \frac{1 - e^{-(2\pi i (k-j)/n) \times n}}{1 - e^{-2\pi i (k-j)/n}} & k \neq j \end{cases} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} .
\end{aligned}$$

So the e_j vectors are an orthonormal basis.

The proof of the orthogonality of the basis is easiest in the complex domain, but visualise what these bases look like I find it is easier to think about the real and imaginary parts separately where we note that

$$\frac{1}{\sqrt{n}} e^{2\pi i k \nu_j} = \frac{1}{\sqrt{n}} \cos(2\pi k \nu_j) + i \frac{1}{\sqrt{n}} \sin(2\pi k \nu_j).$$

Example 6.4.13. (Discrete Fourier Basis) Figure 6.7 shows, for $n = 20$, the real part of the 20 values of e_j for three j values: $3, 5$ and 16. We could have shown the imaginary parts too but they look qualitatively similar.

The 20 values, for each j, are given by the red dots. We have also plotted the underlying, continuous, cosine function for each Fourier frequency value as dashed lines. We see for larger j values these continuous functions have higher frequencies and less observations per cycle.

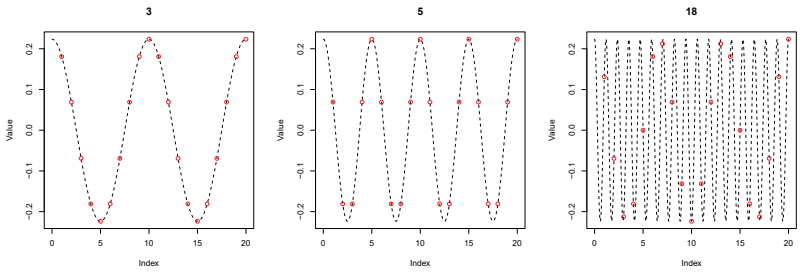

Figure 6.7: Fourier frequencies for $n = 20$: the real (cosine) part of the Fourier frequencies $j = 3, 5, 18$ shown by red dots. The dashed lines are underlying cosine functions.

Definition 6.4.14. (Discrete Fourier transform) If $x \in \mathbb{C}^n$ is a real vector then we can represent it with respect to the bases e_i via the decomposition

$$x = \sum_{j=0}^{n-1} \langle e_j, x \rangle e_j = \sum_{j=0}^{n-1} X(\nu_j) e_j$$

where

$$X(\nu_j) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} x_t e^{-2\pi i \nu_j t}.$$

which is called the j^{th} component of the discrete Fourier transform of x.

Definition 6.4.15. (Periodogram) The periodogram of a real n-vector x with Discrete Fourier transform $(X(\nu_0), X(\nu_1), \ldots, X(\nu_{n-1}))^T$ is the vector with components $I(\nu_j) = |X(\nu_j)|^2$.

This can be written as

$$I(\nu_j) = X_c^2(\nu_j) + X_X^2(\nu_j),$$

where

$$X_c(\nu_j) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \cos(2\pi t \nu_j) x_t, X_s(\nu_j) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \sin(2\pi t \nu_j) x_t$$

We can see the relationship between the periodogram and the sample auto-covariance function, $\widehat{\gamma}$, through the following

Theorem 6.4.16. If we have a real n-vector x, which corresponds to a mean zero stationary process, and its sample acvf is $\widehat{\gamma}$ then

$$I(\nu_j) = \sum_{h=-n+1}^{n-1} \widehat{\gamma}(h) e^{-2\pi i h \nu_j}.$$

Proof. By definition we have trivially that

$$I(\nu_j) = \frac{1}{n}\left|\sum_{t=1}^{n} e^{2\pi i t \nu_j} x_t\right|^2 = \frac{1}{n}\left|\sum_{t=1}^{n} e^{2\pi i t \nu_j}(x_t - \bar{x})\right|^2$$

Now using the result that for a complex number $|z|^2 = z\bar{z}$ we have

$$\begin{aligned}
\frac{1}{n}\left|\sum_{t=1}^{n} e^{2\pi i t \nu_j}(x_t - \bar{x})\right|^2 &= \frac{1}{n}\left(\sum_{t=1}^{n} e^{2\pi i t \nu_j}(x_t - \bar{x})\right)\left(\sum_{t=1}^{n} e^{-2\pi i t \nu_j}(x_t - \bar{x})\right) \\
&= \frac{1}{n}\sum_{s,t} e^{-2\pi i (t-s)\nu_j}(x_t - \bar{x})(x_s - \bar{x}) \\
&= \sum_{h=1-n}^{n-1} \hat{\gamma}(h) e^{-2\pi i h \nu_j}
\end{aligned}$$

since, by definition,

$$\hat{\gamma}(h) := \frac{1}{n}\sum_{t=1}^{n-|h|}\left(x_{t+|h|} - \bar{x}\right)(x_t - \bar{x}),$$

from Definition 3.5.1. □

The theorem gives us a nice link between the auto-covariance function and the periodogram but also points to a problem. We have seen from estimating the auto-covariance function that it is only reliable for $h << n$ – in fact the default of R is to only look at lags smaller than $10\log_{10}(n)$. So this raises the question of whether there is enough information in the data to reliably estimate the spectral density, $f(\nu)$, via the periodogram. The answer is, unfortunately, 'no'. While the periodogram is an unbiased estimate of $f(\nu)$ it does not converge to it as n gets larger. We say that the periodogram is not a consistent estimate of $f(\nu)$.

6.4.4 Smoothing the periodogram

Example (6.4.4 revisited). If we look again at Fig. 6.5, where we simulated from a known spectral distribution, shown in the middle panel. The

right hand panel is the estimated spectral density and we can see that it is very 'rough'. It has approximately the correct shape but is very 'spikey'. This in one way of seeing what it means for the periodogram not be be a consistent estimator of the spectral density. Perhaps it could be improved by smoothing?

Definition 6.4.17. (Daniell kernels) The `kernel` function in R defines different ways of defining a smoother. Here we focus on the Daniell kernel which is a centred moving average. For example, the smoothing formula for a Daniell kernel with $m = 2$ is

$$\hat{x}_t = \frac{x_{t-2} + x_{t-1} + x_t + x_{t+1} + x_{t+2}}{5}$$

The weights for different Daniell kernel's can be formulated using the code

```
> plot(kernel("daniell", c(1)))
> plot(kernel("daniell", c(3,3)))
> plot(kernel("daniell", c(4,4, 4, 4)))
```

The weights which are used in the moving average are shown in Fig. 6.8. We can see they range from a simple equally weighted moving average (left panel) through a triangular distribution (center) to a kernel that looks like a Gaussian one (right).

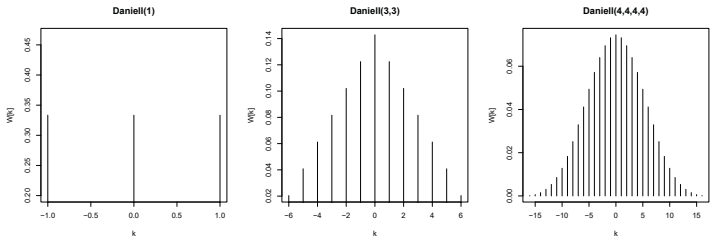

Figure 6.8: Three Daniell kernels

Example (6.4.4 revisited). If we go back to one of the examples in Example 6.4.4 we can see, in Fig. 6.9, the raw (unsmoothed) periodogram and two versions which have been smoothed with a Daniell kernel. The R code is given by

```
> spectrum(out, log="no")
> spectrum(out, log="no", spans= c(3,3))
> spectrum(out, log="no", spans= c(4,4,4,4))
```

Example (1.4.6 revisited). In Chapter 1 we looked at the sunspot data set. The sunspot numbers from 1749 to 1983 were collected at Swiss Federal Observatory, Zurich until 1960, then Tokyo Astronomical Observatory and can be found in R.

R code for this example is given by the following.

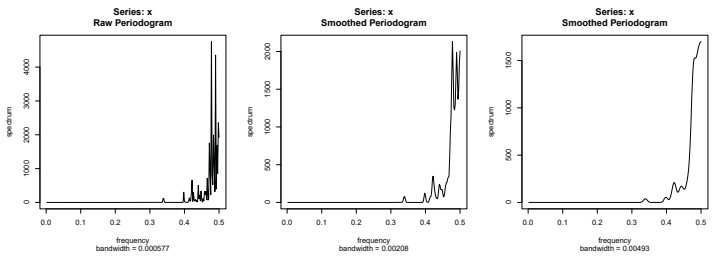

Figure 6.9: The raw periodogram and two smoothed versions

```
> data(sunspots)
> k1 = kernel("daniell", 4)
> plot(sunspots)
> spec.pgram(sunspots,    log = "no", xlim=c(0,1))
> spec.pgram(sunspots, k1,  log = "no", xlim=c(0,1))
```

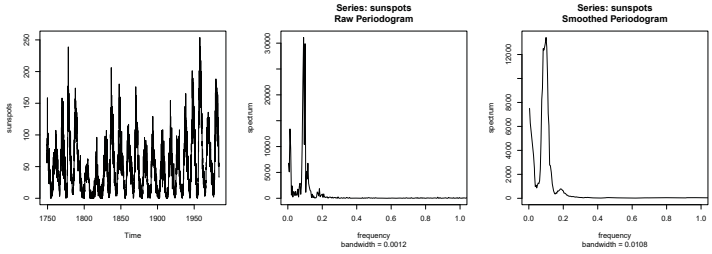

Figure 6.10: The sunspot data, its raw periodogram and a smoothed version

In Fig. 6.10 we see that there is a peak in the spectrum at $\nu = .09$ which corresponds to a $1/0.09 = 11.1$ year cycle.

6.4.5 Filtering in the spectral domain

In this section we see how working in the spectral domain, i.e. working with frequencies, allows us to build tools which filter out 'noise', hopefully leaving 'signal'. The most obvious application of this idea is in signal processing.

Definition 6.4.18. (Signal processing) From Priemer (1991) we have that, for an engineer, a *signal* is a function that conveys information about the behaviour of a system. These can be acoustic waves such as speech or music, electronic waves such as radio signals, or output from a medical device such as EEG. *Signal processing* is then an activity that transforms an input signal to an output signal, typically to amplify the power of the signal.

Example 6.4.19. (Audio signal) For an audio signal – say recorded and transmitted music – noise from the recording and transmission systems –

which sounds like a 'hiss' – often has a different frequency spectrum than the (music) signal. We can therefore separate the 'signal' from the 'noise' by a transformation which has the effect of decreasing power in the high frequency part of the spectrum and increasing power in the parts of the spectrum which corresponds to the music. A filter which does this is called a *low pass filter* since it only allows the lower parts of the spectrum through.

For example Fig. 6.11 shows a simulate example. The left panel shows, in black the raw signal which is a simple sine wave plus i.i.d. noise. Its spectrum, on a log-scale, is shown in the middle panel. We see that the high frequency parts of this spectrum are purely due to the noise. After applying the low pass filter the spectrum changes to the plot in the right panel. The filter has removed some of the high frequency parts. The correpesonding, transformed, signal is shown in red in the left panel. The code which does

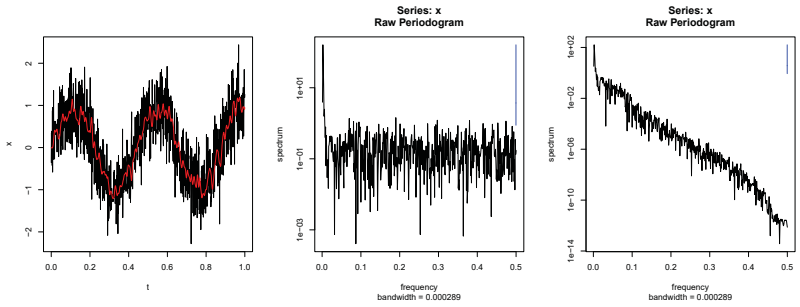

Figure 6.11: Low pass filter example

this in R is given by

```
> library(signal)
> bf <- butter(3, 0.1) # 10 Hz low-pass filter
> t <- seq(0, 1, len = 1000) # 2 second sample
> x <- sin(2*pi*t*2.3) + 0.5*rnorm(length(t)) # 2.3 Hz sinusoid+noise
> z <- filter(bf, x) # apply filter
```

6.5 Appendix

6.5.1 Appendix: Multivariate normal distribution

First recall that we denote linear transformation using

$$y = Ax$$

where x, y are $n \times 1$ (column) vectors and A is $n \times n$ matrix. Further, we say that A is orthogonal if

$$AA^T = I_n$$

where A^T is the transpose of A and I_n the identity matrix

Definition 6.5.1. A *projection* of a vector $x \in R^n$ onto a subspace $S \subset R^n$ is the linear transformation

$$y = Px$$

where P is idempotent i.e. $P.P = P$

In this, and later sections we will make heavy use of the properties of the multivariate normal distribution.

Definition 6.5.2. Let $\boldsymbol{X}$ be a p-dimensional multivariate normal random variable. Then the density of $\boldsymbol{X}$ is given by

$$(2\pi)^{-p/2}|\det \Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\},$$

where $\boldsymbol{\mu} \in \mathrm{R}^p$ is the mean and Σ is the $p \times p$ positive-definite variance-covariance matrix of $\boldsymbol{X}$. The usual notation is to write $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$. Note: $\boldsymbol{x}^T$ means the transpose of $\boldsymbol{x}$.

The properties of the multivariate normal are as follows:

Theorem 6.5.3. Assume assume that $Z \sim N_n(\mu, \Sigma)$. If μ_2 is an n dimensional vector and S a $n \times n$ matrix then

$$SZ + \mu_2 \sim N_n(S\mu + \mu_2, S\Sigma S^T)$$

Note that the dimension does not have to stay the same. For example, A is a $p \times n$ (where $p \leq n$) matrix and $Y = AZ$ then

$$Y \sim N_p(A\mu, A\Sigma A^T)$$

Theorem 6.5.4. Marginal distributions of multivariate normal are again normal. If Y_1 is a $p \times 1$ subvector of a $n \times 1$ vector $Y \sim N_n(\mu, \Sigma)$

Proof. Check formally by defining

$$Y_1 = \left(I_{p\times p}, 0_{p\times(n-p)}\right) Y$$

where I is identity matrix, and 0 is matrix of zeros Then used the linear properties of Normal model to show

$$Y_1 \sim N_p(\mu_1, \Sigma_{11})$$

□

Assume, without loss of generality, that Y_1 is first p components then can write mean and variance covariance as

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{pmatrix}$$

We can also look at the conditional properties of Multivariate normal.

Theorem 6.5.5. The conditional distribution of $Y_1|Y_2 = y_2$ is Normal. It has mean

$$\mu_{1.2} = \mu_1 + \Sigma_{12}(\Sigma_{22})^{-1}(y_2 - \mu_2),$$

and covariance

$$\Sigma_{1.2} = \Sigma_{11} - \Sigma_{12}(\Sigma_{22})^{-1}\Sigma_{12}^T.$$

6.5.2 Appendix: `fkf( )` function and the Kalman filter

In the notation defined by the R function and in §6.5.7 the recursion equations are given explicitly by

```
Let i be the loop variable. The filter iterations are
implemented the following way (in case of no NA's):

Initialization:
if(i == 1){
at[, i] = a0
Pt[,, i] = P0
}

Updating equations:
vt[, i] = yt[, i] - ct[, i] - Zt[,,i] %*% at[, i]
Ft[,, i] = Zt[,, i] %*% Pt[,, i] %*% t(Zt[,, i]) + GGt[,, i]
Kt[,, i] = Pt[,, i] %*% t(Zt[,, i]) %*% solve(Ft[,, i])
att[, i] = at[, i] + Kt[,, i] %*% vt[, i]
Ptt[, i] = Pt[,, i] - Pt[,, i] %*% t(Zt[,, i]) %*% t(Kt[,, i])

Prediction equations:
at[, i + 1] = dt[, i] + Tt[,, i] %*% att[, i]
Pt[,, i + 1] = Tt[,, i] %*% Ptt[,, i] %*% t(Tt[,, i]) + HHt[,, i]

Next iteration:
i <- i + 1
goto 'Updating equations'.
```

6.5.3 Appendix: Bayesian methods and the Kalman filter

From Petris et al. (2009, p. 51) we find the link between the Kalman filter, Bayesian methods and the dynamic linear model through the following theorems.

Theorem 6.5.6. (Filtering recursions)

(i) The one-step state predictive density can be computed from the filtered density by

$$p(\theta_t|\mathcal{D}_{t-1}) = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|\mathcal{D}_{t-1})d\nu(\theta_{t-1})$$

(ii) The one-step observation predictive density can be computed from the state predictive distribution by

$$f(y_t|\mathcal{D}_{t-1}) = \int p(y_t|\theta_t)p(\theta_t|\mathcal{D}_{t-1})d\nu(\theta_t)$$

(iii) The filtering density can be computed from

$$p(\theta_t|\mathcal{D}_t) = \frac{f(y_t|\theta_t)f(\theta_t|\mathcal{D}_{t-1})}{f(y_t|\mathcal{D}_{t-1})}$$

Since the DLM are linear functions of Normal random variables we get the following important results

Theorem 6.5.7. (Kalman filter) For the DLM model in Def. 5.3.4 if

$$\theta_{t-1}|\mathcal{D}_{t-1} \sim N(m_{t-1}, C_{t-1}),$$

for $t > 1$, then

(i) The one-step predictive density of θ_t given $\mathcal{D}_{t-1}$ is normal with

$$a_t := E(\theta_t|\mathcal{D}_{t-1}) = G_t m_{t-1}, R_t := Var(\theta_t|\mathcal{D}_{t-1}) = G_t C_{t-1} G_t^T + W_t$$

(ii) The one-step predictive density of Y_t given $\mathcal{D}_{t-1}$ is normal with

$$f_t := E(Y_t|\mathcal{D}_{t-1}) = F_t a_t, Q_t := Var(Y_t|\mathcal{D}_{t-1}) = F_t^T R_t F_t^T + V_t$$

(iii) The filtering density of θ_t given $\mathcal{D}_t$ is normal with

$$m_t := E(\theta_t|\mathcal{D}_t) = a_t + R_t F_t^T Q_t^{-1} e_t, C_t := Var(\theta_t|\mathcal{D}_t) = R_t - R_t F_t' Q_t^{-1} F_t R_t$$

where $e_t : -Y_t - f_t$ is the forecast error.

We can use the Kalman filter, rather like exponential smoothing, to efficiently compute filtered values and predictions. We start from initial values $\theta_0|\mathcal{D}_0 \sim N(m_0, C_0)$ and compute all the terms recursively, as data becomes available.

One of the very appealing aspects of the Kalman filter its that it can use used form many 'real time' applications since well a new data point become available you only need to compute the next step of the filter – you don't have to repeat any calculations with old data.

Definition 6.5.8. (Gain matrix) The *Gain matrix* is defined as

$$K_t := R_t F_t^T Q_t^{-1}.$$

It measures weight of information used in the 'error correcting' part of the filer.

Bibliography

Abraham, B. and J. Ledolter (2006). *Introduction to regression modeling*. Thomson Brooks/Cole.

Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Bollerslev, T. (2008). Glossary to arch (garch). *CREATES Research Paper 49*.

Bollerslev, T. and E. Ghysels (1996). Periodic autoregressive conditional heteroscedasticity. *Journal of Business & Economic Statistics 14*(2), 139–151.

Box, G. E. and G. M. Jenkins (1976). *Time series analysis: forecasting and control, revised ed*. Holden-Day.

Brockwell, P. J. and R. A. Davis (2002). *Introduction to time series and forecasting*. Springer.

Brockwell, P. J. and R. A. Davis (2009). *Time series: theory and methods*. Springer Science & Business Media.

Brooks, S. P. and G. O. Roberts (1998). Assessing convergence of markov chain monte carlo algorithms. *Statistics and Computing 8*(4), 319–335.

Chatfield, C. and D. Prothero (1973). Box-jenkins seasonal forecasting: problems in a case-study. *Journal of the Royal Statistical Society. Series A (General)*, 295–336.

Cowles, M. K. and B. P. Carlin (1996). Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association 91*(434), 883–904.

Durbin, J. and S. J. Koopman (2012). *Time series analysis by state space methods*. Number 38. Oxford University Press.

Francq, C. and J.-M. Zakoian (2011). *GARCH models: structure, statistical inference and financial applications*. John Wiley & Sons.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2014). *Bayesian data analysis*, Volume 2. Taylor & Francis.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter. (1996). markov chain monte carlo in practice.

Gonzalez-Rivera, G. (2013). *Forecasting for Economics and Business*. The Pearson Series in Economics.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning*, Volume 2. Springer.

Jackman, S. (2009). *Bayesian analysis for the social sciences*, Volume 846. John Wiley & Sons.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering 82*(1), 35–45.

Lamigueiro, O. P. (2014). *Displaying Time Series, Spatial, and Space-time Data with R*. CRC Press.

Linstone, H. A., M. Turoff, et al. (1975). *The Delphi method: Techniques and applications*, Volume 29. Addison-Wesley Reading, MA.

Lunn, D., C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter (2012). *The BUGS Book - A Practical Introduction to Bayesian Analysis*. CRC Press / Chapman and Hall.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979). *Multivariate analysis*. Academic press.

Park, T. and G. Casella (2008). The bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Petris, G., S. Petrone, and P. Campagnoli (2009). *Dynamic linear models with R*. Springer Science & Business Media.

Plummer, M. (2010). Jags: A program for analysis of bayesian graphical models.

Priemer, R. (1991). *Introductory signal processing*, Volume 6. World Scientific.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Sherman, H. J. and D. Kolk (1996). *Business cycles and forecasting*. HarperCollins College Publishers.

Shumway, R. H. and D. S. Stoffer (2010). *Time series analysis and its applications: with R examples*. Springer Science & Business Media.

Sorenson, H. W. (1970). Least-squares estimation: from gauss to kalman. *Spectrum, IEEE 7*(7), 63–68.

Stamey, T. A., J. N. Kabalin, J. E. McNeal, I. M. Johnstone, F. Freiha, E. A. Redwine, and N. Yang (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of urology 141*(5), 1076–1083.

Tong, H. (1990). *Non-linear time series: a dynamical system approach*. Oxford University Press.

Van der Heijden, K. (2011). *Scenarios: the art of strategic conversation*. John Wiley & Sons.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of financial economics 5*(2), 177–188.

West, M. and J. Harrison (1997). *Bayesian forecasting and dynamic models*, Volume 18. Springer New York.

Index