12

State Space Models

12.1 Purpose

The state space formulation for time series models is quite general and encompasses most of the models we have considered so far. However, it is usually simpler to use the specific time series models we have already introduced when they are appropriate for the physical situation. Here, we shall focus on applications for which we require parameters to adapt over time, and to do so more quickly than in a Holt-Winters model. The recent turmoil on the world's stock exchanges[1] is a dramatic reminder that time series are subject to sudden changes. Another desirable feature of state space models is that they can incorporate time series of predictor variables in a straightforward manner.

Control engineers have used a state space representation of physical systems as input, state, and output variables related by first-order linear differential equations since the 1950s, and Kalman and Bucy published their famous paper on filtering in 1961 (Kalman and Bucy, 1961). Plackett (1950) published related, but less general, work on the adaptive estimation of coefficients in regression models and gave some historical background to the problem. In the control context, the state variables define the dynamics of some physical system and might, for instance, be displacements and velocities. Typically, only some of these state variables can be measured directly, and these measurements are subject to noise. The objective of the Kalman filter is to infer values for all the state variables from the noisy measurements. The estimated values of the state variables are then used to control the system. Feedback control systems are the essence of robotics, and some applications are cruise control in automobiles, autopilots in aircraft, and the planetary explorer Rover Sojourner – the Mars Pathfinder Mission was launched on the December 4, 1996.

[1] Notable financial events in 2008 included the US government takeover of Fannie Mae and Freddie Mac on September 7, the rejection of the first bailout bill by the US House of Representatives on September 29, and the passing of the US Emergency Economic Stabilization Act of 2008 on October 3.

P.S.P. Cowpertwait and A.V. Metcalfe, *Introductory Time Series with R*, Use R, DOI 10.1007/978-0-387-88698-5_12,

The chemical process industry provides many other applications for control engineers. Typically, states will be concentrations, temperatures, and pressures, and the controller will actuate burners, stirrers, and pumps. Digital computers are an essential feature of modern control systems, and discrete-time models tend to be used in place of continuous-time models, with differences replacing derivatives and time series replacing continuous (analog) signals.

In this chapter, we focus on economic time series. Usually, the states will be unknown coefficients in the linear models and the equations that represent changes in states will be rather simple. Nevertheless, the concept of such parameters changing rather than being fixed is a departure from most of the models we have considered so far, the exception being the Holt-Winters forecasting method. A Bayesian approach is ideal for the development of a state space model.

12.2 Linear state space models

12.2.1 Dynamic linear model

We adopt the notation used in Pole et al. (1994), who refer to state space models as dynamic linear models. The values of the state at time t are represented by a column matrix θ_t and are a linear combination of the values of the state at time $t-1$ and random variation (system noise) from a multivariate normal distribution. The linear combination of values of the state at time $t-1$ is defined by G_t, and the variance-covariance matrix of the system noise is W_t. The observation at time t is denoted by a column matrix y_t that is a linear combination of the states, determined by a matrix F_t, and random variation (measurement noise) from a normal distribution with variance-covariance matrix V_t. The random variation has mean zero and is uncorrelated over time. All the matrices can be time varying, but in many applications G_t is constant. The state space model is summarised by the equations

$$\begin{aligned} y_t &= F_t'\theta_t + v_t \\ \theta_t &= G_t\theta_{t-1} + w_t \end{aligned} \tag{12.1}$$

where $\theta_0 \sim N(m_0, C_0)$, $v_t \sim N(0, V_t)$, and $w_t \sim N(0, W_t)$.

A specific, but very useful, application of state space models is to generalise regression models so that the parameters can vary over time. For example, the sales manager in a house building company might use the following model to allow for the influence of a general level (L) of sales in the sector and the company's own pricing (P) policy on the company's house sales (S):

$$\begin{aligned} S_t &= L_t + \beta_t P_t + v_t \\ L_t &= L_{t-1} + \Delta L_t \\ \beta_t &= \beta_{t-1} + \Delta\beta_t \end{aligned} \tag{12.2}$$

The first equation is a linear regression with price as the predictor variable. However, the model allows the intercept term, the level, and the coefficient of price to vary over time, and this makes it far more realistic for the house building market. The v_t, ΔL_t, and $\Delta\beta_t$ are random deviations with mean zero that are independent over time, although ΔL_t and $\Delta\beta_t$ can be correlated. The relative magnitudes of the variances of these components of error, which are the entries in the matrices V_t and W_t, determine the variability of the parameters. If $W_t = 0$, the state space model reduces to the standard regression with constant parameters. In state space form

$$y_t = S_t \qquad \theta_t = \begin{pmatrix} L_t \\ \beta_t \end{pmatrix} \qquad w_t = \begin{pmatrix} \Delta L_t \\ \Delta\beta_t \end{pmatrix} \qquad F_t = \begin{pmatrix} 1 \\ P_t \end{pmatrix} \qquad G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The subscript t on the matrix G is redundant, as G is constant in this application.

The system is said to be observable if it is possible to infer the values of all the components of the state from the noisy observations (Exercise 5). If the system is observable, we can distinguish prediction, filtering, and smoothing. Prediction is the forecasting of future values of the state, filtering is making the best estimate of the current values of the state from the record of observations, including the current observation, and smoothing is making the best estimates of past values of the state given the record of observations. Filtering is particularly important because it is the basis for control algorithms and forecasting.

12.2.2 Filtering*

Let D_t represent the data up until time t. In most applications, the data are the time series of observations, but the notation does allow for the time series to be augmented by any additional information. In the following, we express the data up until time t, D_t, as the combination of data up until time $t-1$ and the observation at time t, (D_{t-1}, y_t). Bayes's Theorem gives

$$p(\theta_t \mid D_{t-1}, y_t) = \frac{p(y_t \mid \theta_t)\, p(\theta_t \mid D_{t-1})}{p(y_t)} \tag{12.3}$$

and is usually applied without the normalising constant, so that we can write

$$p(\theta_t \mid D_{t-1}, y_t) \propto p(y_t \mid \theta_t)\, p(\theta_t \mid D_{t-1}) \tag{12.4}$$

That is, the posterior density of the state at time t, given data up until time t, is proportional to the product of the probability density of the observation at time t given the state at time t, referred to as the likelihood, and the prior density of the state at time t, given data up until time $t-1$. If the prior distribution and the likelihood are both normal, Bayes's Theorem provides a nice analytic form. In the univariate case, the mean of the posterior distribution is a weighted mean of the mean of the prior distribution and the observation with weights proportional to their precisions.[2] Also, the precision of the posterior distribution is the sum of the precision of the prior distribution and the precision of the observation (Exercise 6). The extension of this result to the multivariate normal distribution leads to the result

$$\theta_t \mid D_t \sim N(m_t, C_t) \tag{12.5}$$

where m_t and C_t are calculated iteratively, for t from 1 up to n, from the following algorithm, which is known as the Kalman filter. Remember that m_0 and C_0 are specified as part of the model.

Kalman filter

The prior distribution for θ_t, the likelihood, and the posterior distribution for θ_t are (respectively)

$$\theta_t \mid D_{t-1} \sim N(a_t, R_t) \qquad y_t \mid \theta_t \sim N(F_t'\theta_t, V_t) \qquad \theta_t \mid D_t \sim N(m_t, C_t)$$

Then, for $t = 1, \ldots$, the algorithm is given by

$$\begin{aligned} a_t &= G_t m_{t-1} & f_t &= F_t' a_t \\ R_t &= G_t C_{t-1} G_t{}' + W_t & Q_t &= F_t{}' R_t F_t + V_t \\ e_t &= y_t - f_t & A_t &= R_t F_t Q_t^{-1} \\ m_t &= a_t + A_t e_t & C_t &= R_t - A_t Q_t A_t{}' \end{aligned}$$

In this algorithm, f_t is the forecast value of the observation at time t, the forecast being made at time $t-1$. It follows that e_t is the forecast error. The posterior mean is a weighted sum of the prior mean and the forecast error. Notice that the variance of the posterior distribution, C_t, is less than the variance of the prior distribution, R_t.

12.2.3 Prediction*

Predictions start from the posterior estimate of the state, obtained from the Kalman filter, on the day (t) on which the forecast is made. We can make one-step-ahead forecasts in the following manner:

$$\begin{aligned} E\left[y_{t+1} \mid D_t\right] = E\left[F_{t+1}'\theta_{t+1} + v_{t+1} \mid D_t\right] = F_{t+1}' E\left[\theta_{t+1} \mid D_t\right] &= F_{t+1}' a_{t+1} \\ &= f_{t+1} \end{aligned} \tag{12.6}$$

$$\begin{aligned} \mathrm{Var}\left[y_{t+1} \mid D_t\right] = \mathrm{Var}\left[F_{t+1}'\theta_{t+1} + v_{t+1} \mid D_t\right] &= F_{t+1}' \mathrm{Var}\left[\theta_{t+1} \mid D_t\right] F_{t+1} + V_{t+1} \\ &= F_{t+1}' R_{t+1} F_{t+1} + V_{t+1} \\ &= Q_{t+1} \end{aligned} \tag{12.7}$$

[2] Precision is the reciprocal of the variance.

The general formula for a forecast at time t for k steps ahead is

$$y_{t+k} \,|\, D_t \sim N(f_{t+k\,|\,t}, Q_{t+k\,|\,t}) \tag{12.8}$$

where

$$\begin{aligned} f_{t+k\,|\,t} &= F'_{t+k} G^{k-1} a_{t+1} \\ Q_{t+k\,|\,t} &= F'_{t+k} R_{t+k\,|\,t} F_{t+k} + V_{t+k} \\ R_{t+k\,|\,t} &= G^{k-1} R_{t+1} (G^{k-1})' + \textstyle\sum_{j=2}^{k} G^{k-j} W_{t+j} (G^{k-j})' \end{aligned} \tag{12.9}$$

12.2.4 Smoothing*

The optimal smoothing algorithm follows from a nice application of elementary probability. We demonstrate this for one step back, and the general case proceeds in the same way. To begin with, we use the rule of total probability

$$p(\theta_{t-1} \,|\, D_t) = \int p(\theta_{t-1} \,|\, \theta_t, D_t)\, p(\theta_t \,|\, D_t)\, d\theta_t \tag{12.10}$$

The $p(\theta_t \,|\, D_t)$ in the integrand on the right-hand side is available from the Kalman filter, so we only need to consider further

$$p(\theta_{t-1} \,|\, \theta_t, D_t) = p(\theta_{t-1} \,|\, \theta_t, D_{t-1}) \tag{12.11}$$

because y_t provides no further information once θ_t is known. Now we can apply Bayes's Theorem:

$$p(\theta_{t-1} \,|\, \theta_t, D_{t-1}) = \frac{p(\theta_t \,|\, \theta_{t-1}, D_{t-1})\, p(\theta_{t-1} \,|\, D_{t-1})}{p(\theta_t \,|\, D_{t-1})} \tag{12.12}$$

Finally, given θ_t and D_{t-1}, the denominator on the right-hand side is the normalising factor, the first term in the numerator on the right-hand side follows from the system equations, and the second term in the numerator is the posterior density at time $t-1$, which follows from the Kalman filter. If we now assume the distributions are normal we obtain the result

$$\theta_{t-1} \,|\, D_t \sim N(a_t(-1), R_t(-1)) \tag{12.13}$$

where

$$\begin{aligned} a_t(-1) &= m_{t-1} + B_{t-1}(m_t - a_t) \\ R_t(-1) &= C_{t-1} - B_{t-1}(R_t - C_t) B'_{t-1} \\ B_{t-1} &= C_{t-1} G' R_t^{-1} \end{aligned} \tag{12.14}$$

You can find more details in Pole et al. (1994).

12.3 Fitting to simulated univariate time series

12.3.1 Random walk plus noise model

As a first example, consider a daily stock price taken at the close of each trading day. This is treated as independent normal random variation, with a standard deviation of 1 about a mean that is 20 for the first 10 time points but drops to 10 for time points 11 up to 20. In practice, we never know the underlying process, and models we fit are based on physical intuition and the goodness-of-fit to the data. State space models have the great advantage that the parameters can change over time and are able to allow for a change in mean level. We first implement a model for the stock price, y_t,

$$\begin{aligned} y_t &= \theta_t + v_t \\ \theta_t &= \theta_{t-1} + w_t \end{aligned} \tag{12.15}$$

where $\theta_0 \sim N(25, 10)$, $v_t \sim N(0, 2)$, and $w_t \sim N(0, 0.1)$, which allows for small changes in an underlying mean level θ_t. The SS function in the sspir package (Dethlefsen and Lundbye-Christensen, 2006) creates a state space object.[3] The syntax corresponds precisely to the notation of Equation 12.1 except for the additional phi parameter, which we do not use in this chapter and can be ignored. The function kfilter gives the Kalman filter estimate of the state at each time point, given the preceding observations and the observation at that time. The function smooth gives the retrospective estimate of the state at each time given the entire time series.

```
> library(sspir)
> set.seed(1)
> Plummet.dat <- 20 + 2*rnorm(20) + c(rep(0,10), rep(-10,10))
> n <- length(Plummet.dat)
> Plummet.mat <- matrix(Plummet.dat, nrow = n, ncol = 1)
> m1 <- SS(y = Plummet.mat,
           Fmat = function(tt,x,phi) return( matrix(1) ),
           Gmat = function(tt,x,phi) return( matrix(1) ),
           Wmat = function(tt,x,phi) return( matrix(0.1)),
           Vmat = function(tt,x,phi) return( matrix(2) ),
           m0   = matrix(25), C0 = matrix(10)
          )
> plot(m1$y, ylab = "Closing price", main = "Simulated")
> m1.f <- kfilter(m1)
> m1.s <- smoother(m1.f)
> lines(m1.f$m, lty = 2)
> lines(m1.s$m, lty = 3)
```

In Figure 12.1, the Kalman filter rapidly settles around 20 because of the relatively high variance of 10 that we have attributed to our initial inaccurate

[3] You will need to download this package from CRAN.

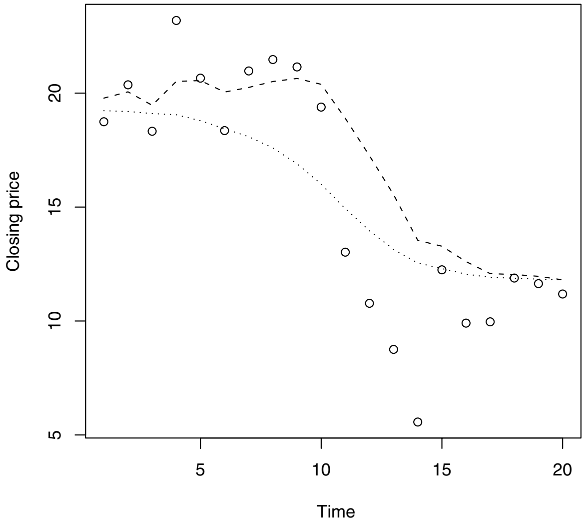

Fig. 12.1. Simulated stock closing prices: filtered estimate of mean level (dashed); smoothed estimate of mean level (dotted).

estimate of 25 until time 10. However, the filter is slower to adapt to the step change. We can improve this performance, if we take advantage of the Bayesian formulation, which is ideally suited for incorporating our latest information. Given the drop in mean level at $t = 11$, it would be prudent to review our assessment of the relevance of the earlier time series to future values. Here, as an example, we decide to assign a variance of 10 to the evolution of θ_t at $t = 12$. This effectively allows the filter to restart at $t = 12$, and in Figure 12.2 you can see that the estimate of θ_t rapidly settles about 10 after $t = 12$. For comparison, the filter without intervention is also shown in Figure 12.2. If you look back at Figure 12.1, you will see the smoothed estimate of θ_t. In this case, it gives less accurate estimates of θ_t than when $t < 20$ because the assumed model, without intervention, makes little allowance for a large step change. In any application, the filter and smoother must coincide for the latest observation. The latest filtered value is our best estimate of the mean level and, for this model, our best estimate of tomorrow's price.

Another means of making the filter adapt more quickly to a change in level is to increase the variance of w_t. The drawback is that the filter will then be unduly influenced by daily fluctuations in price if the level is constant. It is the ratio of the variance of w_t to the variance of v_t rather than the absolute values of the variances that determines the filter path (Exercise 1). However, limits of prediction do depend on the absolute values.

```
plot(m1$y, ylab = "Closing price", main = "Simulated")
m1.f <- kfilter(m1)
lines(m1.f$m, lty = 2)
m2 <- m1
Wmat(m2) <- function(tt, x, phi) {
              if (tt == 12) return(matrix(10)) else return(matrix(0.1))
}

m2.f <- kfilter(m2)
lines(m2.f$m,lty=4)
```

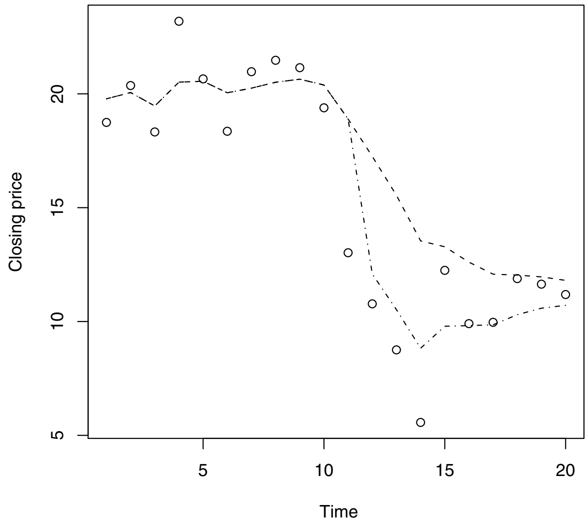

Fig. 12.2. Simulated stock closing prices: filter after intervention (dot-dash); original filter (dashed).

In bull markets, stock prices tend to drift upwards. A drift can be incorporated in the state space model by introducing an additional element in the state vector. You are asked to do this in Exercise 2.

12.3.2 Regression model with time-varying coefficients

The time series regression models that we considered in Chapter 5 are based on an assumption that the process is stationary and hence that the coefficients are constant. This assumption is particularly equivocal for recent environmental and economic time series, even if the predictor variables do not explicitly include time, and state space models are ideally suited to relaxing the restriction. To demonstrate the procedure, we will generate data from the equations

$$\begin{aligned} y_t &= a + bx_t + z_t \\ x_t &= 2 + t/10 \end{aligned} \tag{12.16}$$

where $t = 1, \ldots, 30$, $z_t \sim N(0, 1)$, $a = 4$ and $b = 2$ for $t = 1, \ldots, 15$, and $a = 5$ and $b = -1$ for $t = 16, \ldots, 30$. We fit a straight line with time-varying coefficients, which is the model recommended to the sales manager in the house building company of §12.2.1. The components of θ_t are the intercept and slope at time t, and the estimates from the Kalman filter are shown in Figure 12.3. In this application, the matrix F, `Fmat` in the `SS` function, is time varying and is $(1, x_t)'$. The matrix `Gmat` is the identity matrix, which we use `diag` to generate. The parameter variation, which is modelled with matrix `Wmat`, is small relative to the observation variance modelled by `Vmat`. The initial guesses for the intercept and slope are 5 and 3, respectively, and the associated variance of 10 reflects the considerable uncertainty.

```
> library(sspir)
> set.seed(1)
> x1 <- 1:30
> x1 <- x1/10 + 2
> a <- c(rep(4,15), rep(5,15))
> b <- c(rep(2,15), rep(-1,15))
> n <- length(x1)
> y1 <- a + b * x1 + rnorm(n)
> x0 <- rep(1, n)
> xx <- cbind(x0, x1)
> x.mat <- matrix(xx, nrow = n, ncol = 2)
> y.mat <- matrix(y1, nrow = n, ncol = 1)

> m1 <- SS(y = y.mat, x = x.mat,
          Fmat = function(tt,x,phi)
            return( matrix(c(x[tt,1], x[tt,2]), nrow = 2, ncol = 1)),
          Gmat = function(tt,x,phi) return (diag(2)),
          Wmat = function(tt, x, phi) return (0.1*diag(2)),
          Vmat = function(tt,x,phi) return (matrix(1)),
          m0 = matrix(c(5,3),nrow=1,ncol=2),C0=10*diag(2)
          )

> m1.f <- kfilter(m1)
> par(mfcol=c(2,1))
> plot(m1.f$m[,1], type='l')
> plot(m1.f$m[,2], type='l')
```

The estimates from the Kalman filter rapidly approach the known values, even after the step change (Fig. 12.3). The estimated standard errors of the estimates from the filter are based on the specified values in the variance matrices. rather than being estimated from the data (§12.7). Pole et al. (1994) give further details of state space models with estimated variances.

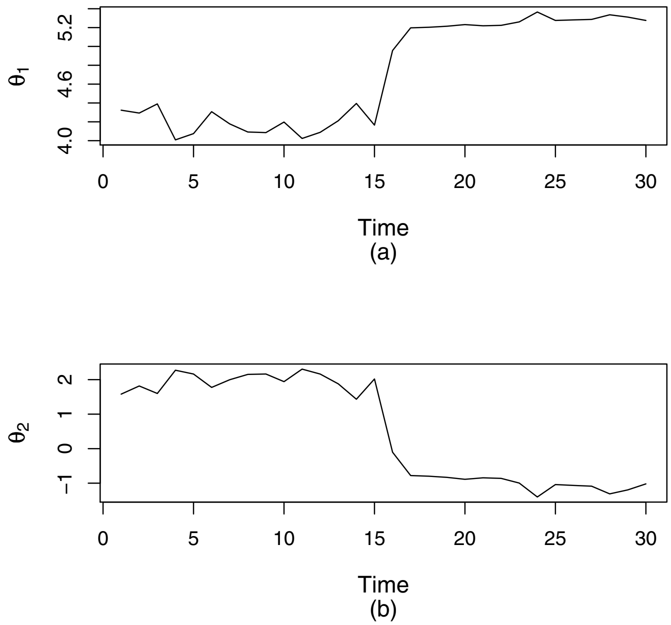

Fig. 12.3. Realisation of a regression model in which the intercept and slope change. Kalman filter estimates of (a) intercept and (b) slope.

12.4 Fitting to univariate time series

Morgan Stanley share prices for trading days from Monday, August 18, 2008, until Friday, November 14, 2008, are in the online file `MorgStan.dat`. If we wish to set up a random walk plus drift model, we need an estimate of the two variance components. One way of doing this is to compare the variances within and between weeks, the former being taken as V and the latter as W (Fig. 12.4). Late 2008 was a tumultuous period for bank shares and the variances within and between weeks are estimated as 5.4 and 106, respectively (Exercise 3). With these parameter values, both the filtered and smoothed values are very close to the observed data. The estimated price of shares on Monday, November 17, is given by the latest filtered value `m1.f$m[n]` and equals 12.08, which is close to the 12.03 closing price on Friday, November

14. If the variances within and between weeks are estimated from the first four weeks, when the market was relatively stable, they are 2.1 and 1.0. The estimated price of shares on Monday, November 17, is now 12.69.

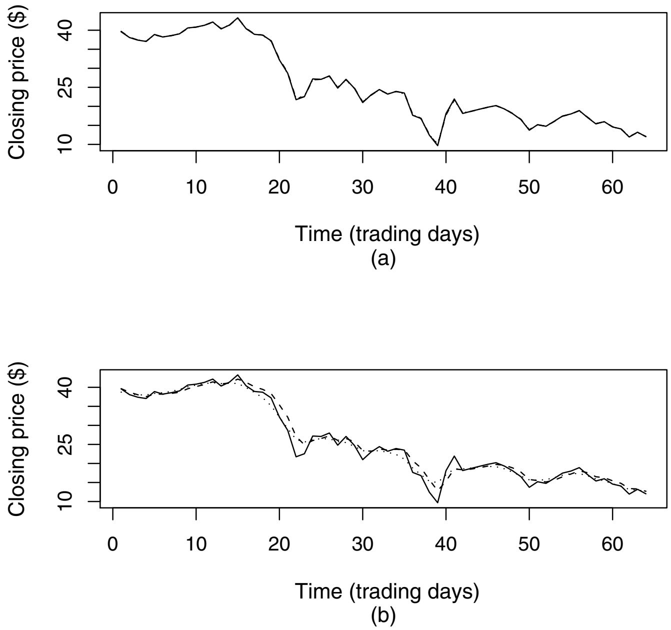

Fig. 12.4. Morgan Stanley close of business share prices for trading days August 18 until November 14, 2008. Kalman filter and smoothed values: (a) $V = 5.4$ and $W = 106$; (b) $V = 2.2$ and $W = 1.0$.

12.5 Bivariate time series – river salinity

The Murray River supplies about half of South Australia's urban water needs and, in dry years, this can increase to as much as ninety percent. Other sources of water in South Australia are bore water and recycled water, although both tend to have high salinity. The World Health Organisation (WHO) recommendation for the upper limit of salinity of potable water is 500 mg/l (approximately 800 EC), but the domestic grey water system and some industrial and irrigation users can tolerate higher levels of salinity. The low rainfall and increasing population makes the efficient use of water resources a priority, and there are water-blending schemes that aim to maximise the use of recycled water. The average monthly salinity, measured by electrical conductivity (EC; microSiemens per centimetre at 25°C) and flow (Gigalitres per month) at

Chowilla on the Murray River, have been calculated from data provided by the Government of South Australia and are available in the file `Murray.txt`. Predictions of salinity are needed for the recycled water schemes to be operated efficiently, and the changing level of salinity requires an adaptive forecasting strategy, which is well supported by state space methods.

We have 81 months of flows and salt concentrations at Chowilla and have been asked to set up a state space model that can adapt to the changing salt and flow levels that are a feature of the Murray River. Let S_t and L_t be the mean adjusted salt concentration and river flow for month t, respectively. The mean adjustment generally improves numerical stability and has as a convenient consequence that the estimates of the intercept terms and estimates of other coefficients will be approximately uncorrelated. A preliminary time series analysis found that the model

$$S_t = \theta_1 + \theta_2 S_{t-1} + \theta_3 L_{t-1} + \theta_4 \cos(2\pi t/12) + \theta_5 \sin(2\pi t/12) + v_{S,t} \quad (12.17)$$

$$L_t = \theta_6 + \theta_7 S_{t-1} + \theta_8 L_{t-1} + \theta_9 \cos(2\pi t/12) + \theta_{10} \sin(2\pi t/12) + v_{L,t} \quad (12.18)$$

provides a good fit to the available data (Exercise 4). We now express this in state space form, which will allow for the coefficients $\theta_1, \ldots, \theta_{10}$ to change over time.

$$\begin{pmatrix} S_t \\ L_t \end{pmatrix} = \begin{pmatrix} 1 & S_{t-1} & L_{t-1} & cs_t & sn_t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & S_{t-1} & L_{t-1} & cs_t & sn_t \end{pmatrix}$$
$$\times \left(\theta_{1,t}\ \theta_{2,t}\ \theta_{3,t}\ \theta_{4,t}\ \theta_{5,t}\ \theta_{6,t}\ \theta_{7,t}\ \theta_{8,t}\ \theta_{9,t}\ \theta_{10,t} \right)' \quad (12.19)$$

$$\theta_{i,t} = \theta_{i,t-1} + w_{i,t} \quad (12.20)$$

The `R` code is now more succinct and makes use of `diag` to set up diagonal matrices, but the general principles are the same as for any regression model with time-varying coefficients. The diagonal elements of the matrix `Vmat` are the estimated variances of the errors from the preliminary regression models for S_t and L_t, which are 839 and 1612, respectively. There is no evidence of autocorrelation in the residual series from the two regressions, but the cross-correlation at lag 0, which is -0.299, is statistically significant. The corresponding estimate of the covariance of the errors, the off-diagonal term in `Vmat`, is therefore -348. The matrix `Wmat` is set up to allow the mean levels to adapt and slight adaptation of the other coefficients. The choice of values for the variances is somewhat subjective. The mean salinity and mean flow over the 81-month period were 165 and 259, respectively, and a variance of 10 corresponds to a standard deviation of roughly 2% of mean salinity and 1% of mean flow. The variances of 0.0001 correspond to a standard deviation of 0.01 for the change in level of the other coefficients. The effect of changing entries in `Wmat` can be investigated when setting up the filter. The initial estimates in `m0` were set fairly close to the estimates from the preliminary regressions. The uncertainty associated with these estimates was arbitrarily set at 100 times Wmat, as it does not have a critical effect on the performance of the filter. The results are shown in Figures 12.5 and 12.6 and seem reasonable.

```
> library(sspir)
> www <- 'http://www.massey.ac.nz/~pscowper/ts/Murray.txt'
> Salt.dat <- read.table(www, header=T) ; attach(Salt.dat)
> n <- 81 ; Time <- 1:n
> SIN  <- sin(2 * pi * Time /12)[-1]
> COS  <- cos(2 * pi * Time /12)[-1]
> Chowilla <- Chowilla - mean(Chowilla)
> Flow <- Flow - mean(Flow)
> Chow <- Chowilla[-1]
> Chow.L1 <- Chowilla[-n]
> Flo <- Flow[-1]
> Flo.L1 <- Flow[-n]
> Sal.mat <- matrix(c(Chow, Flo), nrow = 80, ncol = 2)
> x0 <- rep(1, (n-1))
> xx <- cbind(x0, Chow.L1, Flo.L1, COS, SIN)
> x.mat <- matrix(xx, nrow = n-1, ncol = 5)
> G.mat <- diag(10)
> W.mat <- diag(rep(c(10, 0.0001, 0.0001, 0.0001, 0.0001), 2))
> m1 <- SS(y = Sal.mat, x = x.mat,
         Fmat =
          function(tt, x, phi) return (matrix(
            c(x[tt,1], x[tt,2], x[tt,3], x[tt,4], x[tt,5], rep(0,10),
              x[tt,1], x[tt,2], x[tt,3], x[tt,4], x[tt,5]),
                                nrow=10,ncol=2)),
         Gmat = function(tt, x, phi) return (G.mat),
         Wmat = function(tt, x, phi) return (W.mat),
         Vmat = function(tt, x, phi) return
                  (matrix(c(839, -348, -348, 1612), nrow=2, ncol=2)),
         m0=matrix(c(0,0.9,0.1,-15,-10,0,0,0.7,30,20),nrow=1,ncol=10),
         C0 = 100 * W.mat
         )

> m1.f <- kfilter (m1)
> par(mfcol=c(2,3))
> plot(m1.f$m[,1], type='l')
> plot(m1.f$m[,2], type='l')
> plot(m1.f$m[,3], type='l')
> plot(m1.f$m[,6], type='l')
> plot(m1.f$m[,7], type='l')
> plot(m1.f$m[,8], type='l')
>
> par(mfcol=c(2,2))
> plot(m1.f$m[,4], type='l')
> plot(m1.f$m[,5], type='l')
```

```
> plot(m1.f$m[,9], type='l')
> plot(m1.f$m[,10], type='l')
```

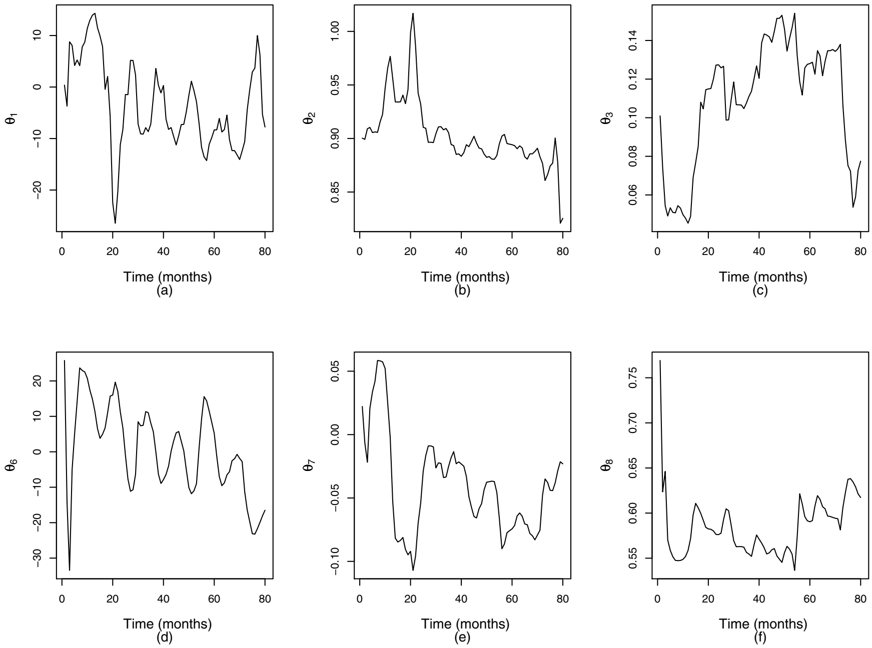

Fig. 12.5. Kalman filter estimates of parameters for the Murray River salt and flow model: (a) deviation of salt from mean; (b) autoregressive coefficient for salt; (c) cross regressive coefficient for flow; (d) deviation of flow from mean; (e) cross regressive coefficient for salt; (f) autoregressive coefficient for flow.

12.6 Estimating the variance matrices

In the applications considered in this chapter, we have had to specify values for the V and W matrices when setting up the state space model. We can adapt the algorithm to update these values as we obtain more data. The formula for the variance of one-step-ahead forecast errors depends on the known values of entries in the V and W matrices. This theoretical variance can be compared with the variance of the actual forecast errors up until time t. Define

$$\phi_t = \frac{\text{actual variance of forecast errors up to time } t}{\text{theoretical variance of forecast errors}}$$

Then updated matrices for V and W are obtained from $\phi_t V$ and $\phi_t W$, respectively. This strategy will not adjust the relative variances of measurement noise and system noise, but it does allow the absolute values to be updated (Exercise 7).

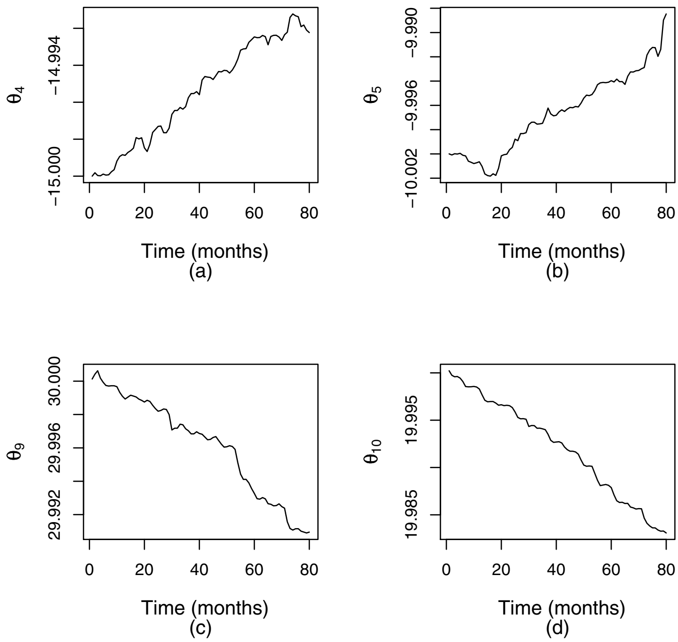

Fig. 12.6. Kalman filter estimates of coefficients of seasonal components for Murray River salt and flow model: (a) cosine for salt; (b) sine for salt; (c) cosine for flow; (d) sine for flow.

12.7 Discussion

One of the main advantages of state space models is that they are adaptive, and the benefits of this are usually realised by implementing them in real time. We have only covered relatively straightforward examples, and there are many useful extensions. In particular, there are sophisticated methods for estimating the variances rather than specifying them and methods for estimating parameters in the F and G matrices as well as the states when this is theoretically possible. The distinction between states and unknown parameters depends on the application (see Exercise 5).

The Kalman filter applies to linear systems, but it can also be used as a local linear approximation to a non-linear system. This important development is known as the extended Kalman filter. The optimality of the standard Kalman filter rests on an assumption that the noise is Gaussian (normal) and independent, but similar optimal filters have been developed with other assumptions about noise distributions.

12.8 Summary of additional commands used

`SS`	Sets up state space model
`kfilter`	Runs the Kalman filter
`smooth`	Creates smoothed estimates of past values

12.9 Exercises

1. Refer to the `R` script to simulate stock price in §12.3.1.
   a) Change the variance of w_t from 0.1 to 10, and comment on the change in the filter and smoother paths.
   b) Change the variance of w_t from 0.1 to 10 and the variance of v_t from 2 to 200, and comment on the change in the filter and smoother paths.

2. Suggest a means of incorporating a drift in the random walk plus noise model by introducing a second element in the state θ_t. Apply your model to the Morgan Stanley share price series.

3. The 64 Morgan Stanley share prices are from trading days over 13 weeks, Monday, September 1, being the Labor Day holiday.
   a) Calculate the variance for each of the 13 weeks, all but the third consisting of five trading days. This is the estimate of the within-week variance S^2_{within}.
   b) Calculate the mean for each of the weeks and calculate the variance of the 13 means. Denote this by $S^2_{\bar{x}}$.
   c) Estimate the variance between weeks as

   $$S^2_{\text{between}} = S^2_{\bar{x}} - \frac{1}{5} S^2_{\text{within}}$$

   The slight inaccuracy that results from one week having only four trading days is negligible.
   d) If you are familiar with the analysis of variance, use `aov` to obtain estimates of the within-week variances that do allow for the Labor Day holiday.

4. Calculate the preliminary regressions for the Murray River salinity example, and verify the numbers given in the text.

5. In many control applications, the matrices F_t and G_t are constant. The first issue is whether or not all the values of the state at time t can be inferred from the observations at time t. If they can be inferred, the system is said to be observable. The linear system in Equation (12.1) is observable if the observability matrix, O, defined by

$$O = \left(F' \; F'G \cdots \; F'G^{p-1} \right)'$$

has full rank, p. Consider a state space model for mean adjusted salinity (S_t) and flow (L_t) in which the coefficients are assumed known and the salinity and flow are the components of the state. Suppose only flow is measured, and assume the model has the form

$$L_t = (0 \quad 1)\,(S_t \quad L_t)' + v_t$$
$$\begin{pmatrix} S_t \\ L_t \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} S_{t-1} \\ L_{t-1} \end{pmatrix} + \begin{pmatrix} w_{S,t} \\ w_{L,t} \end{pmatrix}$$

Can the salinity at time t be inferred from the flow measurement? If so, what are the conditions on a, b, c, and d? (In time series applications, F_t will generally contain the values of the predictor variables at time t. The observability requirement at time t is that the $t \times 1$ columns of predictors in the linear regression model be linearly independent.)

6. Suppose that an observation y has a normal distribution with mean θ and variance ϕ and that the prior distribution for θ is normal with mean θ_0 and variance ϕ_0. We require the posterior distribution of θ given the observation y. From Bayes's Theorem,

$$\begin{aligned} p(\theta \,|\, y) &\propto p(\theta)\, p(y \,|\, \theta) \\ &\propto \exp\left[-\frac{(\theta - \theta_0)^2}{2\phi_0} \right] \exp\left[-\frac{(y - \theta)^2}{2\phi} \right] \end{aligned}$$

It is now convenient to anticipate the result, that the mean of the posterior distribution (θ_1) is a weighted mean of the mean of the prior distribution and the observation with weights proportional to the precision, and define θ_1 and ϕ_1 as

$$\theta_1 = \frac{\phi_0^{-1}}{\phi_0^{-1} + \phi^{-1}}\, \theta_0 + \frac{\phi^{-1}}{\phi_0^{-1} + \phi^{-1}}\, x$$
$$\phi_1^{-1} = \phi_0^{-1} + \phi^{-1}$$

Now use these expressions to replace θ_0 and ϕ_0 in the expression for $p(\theta \,|\, y)$ by θ_1 and ϕ_1:

$$p(\theta \,|\, y) \propto \exp\left[-\frac{\theta^2 + 2\theta\theta_1}{2\phi_1} \right]$$

This is proportional to a normal distribution with mean θ_1 and variance ϕ_1 since θ_1^2/ϕ_1 is a constant with respect to the prior distribution. So,

$$p(\theta, y) = \frac{1}{\sqrt{2\pi\phi_1}} \exp\left[-\frac{(\theta - \theta_1)^2}{2\phi_1} \right]$$

The extension of this result to the multivariate normal distribution can be used to derive the Kalman filter.

7. Verify that the following script is a Kalman filter. Compare its performance with `kfilter` on the simulated regression example. Adapt the filter to update the variance and investigate its performance.

```
> set.seed(1)
> x1 <- c(1:30)
> x1 <- x1/10 + 2
> a <- 4
> b <- 2
> n <- length(x1)
> y1 <- a + b * x1 + 0.1 * rnorm(n)
> x0 <- rep(1, n)
> xx <- cbind(x0, x1)
> F <- matrix(xx, nrow = n,ncol=2)
> y <- matrix(y1, nrow = n,ncol=1)
> G <- matrix(c(1,0,0,1), nrow = 2, ncol = 2)
> W <- matrix(c(1,0,0,1), nrow = 2, ncol = 2)
> V <- matrix(1)
> m0 <- matrix(c(5,1.5), nrow = 2, ncol = 1)
> C0 <- matrix(c(.1,0,0,.1), nrow = 2, ncol = 2)
> a <- 0;R <- 0;f <- 0;Q <- 0;e <- 0;A <- 0;m <- 0;C <- 0;tt <- 0;
> Kfilt.m <- cbind(rep(0, n), rep(0, n))
> m <- m0
> C <- C0
> for (tt in 1:n) {
    Fmat <- matrix(c(F[tt,1],F[tt,2]), nrow = 2, ncol = 1)
    a <- G %*% m
    R <- G %*% C %*% t(G) + W
    f <- t(Fmat) %*% a
    Q <- t(Fmat) %*% R %*% Fmat + V
    e <- y[tt]-f
    A <- R %*% Fmat %*% solve(Q)
    m <- a + A %*% e
    C <- R - A %*% Q %*% t(A)
    Kfilt.m[tt,1] <- m[1,1]
    Kfilt.m[tt,2] <- m[2,1]
  }
> plot(Kfilt.m[1:n, 1])
> plot(Kfilt.m[1:n, 2])
```