State space filtering and smoothing using square root methods

Granville Tunnicliffe Wilson*

August 1, 2017

1 The state space model and state estimation

We will describe the basic computations used to estimate the states of a simple state space model for multivariate time series as presented in Section 2.12 of Tunnicliffe Wilson et al. (2015) and used there to illustrate the prediction of future and other unknown values of a time series. These computations are implemented in the MATLAB function `SSfilsmoMV.m` available on the web site of the book at `http://www.dependenttimeseries.com`.

The model represents an observed m-dimensional time series x_t as linear combinations:

$$x_t = H\, S_t. \tag{1}$$

of an unobserved d-dimensional state vector process S_t. We assume that H is fixed in time, though the model extends without difficulty to time variation of the observation equation (1) including, even, variations in the dimensions m and d.

The process S_t is further modeled as evolving according to the first order Markovian state transition equation:

$$S_t = T\, S_{t-1} + E_t, \tag{2}$$

where T is a fixed square transition matrix and E_t is a d dimensional multivariate white noise disturbance process with constant variance matrix V_E. Again, this equation can be readily generalized to allow time variation of T and V_E.

This is a purely stochastic model. More general state space models include additive terms to represent controllable inputs to the system described by the model.

The value of the state space model and its Markovian structure is that, given observations of $x_1, \ldots, x_n$ it allows efficient sequential computation of the minimum error variance estimates of the sequence of state vectors $S_1, \ldots, S_n$ as linear functions of *present and past* observations $x_1, \ldots, x_t$. These are known as the filtered state estimates.

Further, the minimum error variance estimates of the reversed sequence of state vectors $S_n, \ldots, S_1$ as linear functions of *all* observations $x_1, \ldots, x_n$ can also be efficiently computed in reverse sequence. These are known as the smoothed state estimates. This terminology arises because in many typical applications where the aim is to estimate a signal affected by observation noise, the smoothed estimates appear smoother than the filtered estimates.

*Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK.

In fact the observation equation (1) is commonly written with an additive white noise term representing observation noise. This term can, however, be formally included as a component of the state vector. We will assume that, where relevant, this has been done, mainly because it simplifies our development. It can lead to a slight reduction in computational efficiency, but has some other advantages, such as directly providing an estimate, with error variance, of the observation error term.

Although we refer to the series x_t as observed, a very useful application of state estimation is to situations where some of the components of the series are unknown, possibly over various periods of time, so that these may be estimated (or predicted) from the known terms of the observed series. Such are the applications in Section 2.12 of the book.

Also, we refer to the states as unobserved, but in practice some (or all) of the elements of the state vector at a particular time, may be equated with series values at the same or previous time. Again, such are the applications in Section 2.12 of the book, where estimation of unknown states therefore allows estimation of unknown series values.

A state model is not algebraically unique because a similarity transformation of the state transition by a non-singular matrix M provides an equivalent representation

$$x_t = (H\,M^{-1})U_t = A\,U_t$$

where

$$U_t = M\,S_t = (M\,T\,M^{-1})U_{t-1} + M\,E_t = D\,U_{t-1} + F_t.$$

However, a model is generally formulated so that the states have a particular interpretation which characterizes their definition, and this would be lost by an arbitrary transformation.

There is a property of state space models know as minimality. We will say that a model is *non-minimal* if we can express $x_t = LR_t$ in terms of a new state vector $R_t = NS_t$ of *reduced* dimension that is a linear function of the former state vector. A simple non-minimal model is one in which there are some components of the state state vector which follow a transition that does not affect the other states and upon which the observation x_t is not dependent. These components can then be discarded. More generally, a model is non-minimal if it can be brought to that form by a similarity transformation. A non-minimal model may be acceptable if reducing it to a minimal form loses the useful interpretation of the states. The application of filtering and smoothing to such a model would typically incur some loss of computational efficiency, but no other penalty.

There is however an important numerical concern that may arise in applying filtering and smoothing using the standard method of sequential calculation of the state estimate and its estimation error variance - which, for simplicity, we will simply call the estimation variance. This may arise when the estimate of a state component, or a linear combination of components, becomes exact or nearly so, i.e. the estimation variance becomes, or approaches, zero or singularity. If numerical rounding leads to this variance matrix losing its positive, or non-negative, property, the sequential calculations may become unstable and lead to numerical error of unacceptable magnitude. One way to counter this is to avoid explicit use of this variance matrix by using instead what is known as its square root - a matrix which, if post-multiplied by its transpose, would give the variance matrix. Standard numerical algebraic methods make this feasible with little effect on the speed of computation. This square root method more generally tends to improve numerical accuracy of the computations.

2 Formulation of the square root method

Because of established usage of notation in computational linear algebra we will take the square root of a non-negative symmetric matrix V to be the right factor, i.e. a matrix R that satisfies $R'R = V$.

A practical interpretation of the square root when V is the variance matrix of a vector variable X is that we may express

$$X = R'\,a \tag{3}$$

where a is a vector of uncorrelated variables with identity variance matrix I.

The square root is not unique. If Q is any orthonormal matrix, we also have

$$X = R'\,Q'\,Q\,a = \tilde{R}'\,\tilde{a}$$

so $\tilde{R} = Q\,R$ is also a square root - and any two square roots are related in this way.

One way to constrain the square root to be unique is to require that R is an upper (or right) triangular matrix that satisfies $R'R = V$. In that case we can interpret an element a_i of a in (3) as the error in the minimum variance linear prediction of X_i from $X_1, \ldots X_{i-1}$.

We will use two standard functions in relation to the algebraic manipulation of matrix square roots. The first of these is Choleski factorization. Given a positive definite V this returns R such that $R'\,R = V$ and R is upper triangular. This is applied to construct the square root of the matrix V_E in our application to filtering and smoothing a VAR(p) process. In the state space formulation of this process given in (2.72) of the book, the $d \times d$ matrix V_E, where $d = m\,p$, is zero except for the upper left block matrix which is the model innovation variance V_e. Let R_e be the Choleski factor of V_e. Then the square root of V_E is the $m \times d$ matrix R_E which is zero apart from the leftmost $m \times m$ block which is equal to R_e. So

$$R'_E R_E = \begin{pmatrix} R'_e \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} R_e & 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} V_e & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix} = V_E.$$

In general we will express the error term in the state transition equation (2) as

$$E_t = R'_E\,e_t, \tag{4}$$

where e_t has identity variance matrix.

The second standard function is known as the QR decomposition. For a general $k \times d$ matrix A, this returns a matrix Q with orthonormal columns and an upper triangular matrix R such that $A = Q\,R$. In our applications with $k \geq d$, Q will be the same size as A and R will be square, of size d. Then $A'\,A$ is also square of size d and satisfies

$$A'\,A = R'\,Q'\,Q\,R = R'\,R.$$

3 The recursive step of filtering

This recursion relies on the conditional independence structure of the state space model. Using the language of graphical modeling in Chapter 5 of Tunnicliffe Wilson et al. (2015),

we may represent the model by the directed acyclic graph (DAG) shown in Figure 1(a). This shows the designed causal dependence of each state on the preceding state and of each observation on the contemporaneous state. The conditional independence graph between the same variables, which may be derived from the CIG, is shown in Figure 1(b).

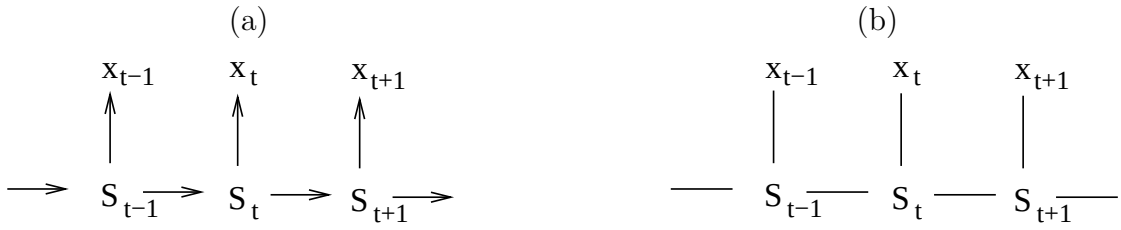

Figure 1: (a) the directed acyclic graph of the state space model and (b) its conditional independence graph.

Let x_{t-} be the set of all past variables $\{x_{t-1}, x_{t-2}, \ldots, x_1\}$ at time t. Writing $P(.)$ for distribution, we wish to derive a forward recursion for $P(S_t|x_t, x_{t-})$ from $P(S_{t-1}|x_{t-})$. The CIG allows us to do this in two steps by exploiting two of its properties:

$$P(S_t|S_{t-1}, x_{t-}) = P(S_t|S_{t-1}) \tag{5}$$

and

$$P(x_t|S_t, x_{t-}) = P(x_t|S_t). \tag{6}$$

Using (5) gives the simplification

$$\begin{aligned} P(S_t|x_{t-}) = \int_{S_{t-1}} P(S_t, S_{t-1}|x_{t-}) &= \int_{S_{t-1}} P(S_t|S_{t-1}, x_{t-}) P(S_{t-1}|x_{t-}) \\ &= \int_{S_{t-1}} P(S_t|S_{t-1}) P(S_{t-1}|x_{t-}), \end{aligned} \tag{7}$$

and using (6) then gives

$$\begin{aligned} P(S_t|x_t, x_{t-}) \propto P(S_t, x_t|x_{t-}) &= P(x_t|S_t, x_{t-}) P(S_t|x_{t-1}) \\ &= P(x_t|S_t) P(S_t|x_{t-}), \end{aligned} \tag{8}$$

The equation (7) is the first step of the recursion, and (8) is the second. For the implementation we will assume that all distributions are Gaussian, but the procedure that results is also valid for linear minimum mean square error prediction.

For the first step we suppose $P(S_{t-1}|x_{t-})$ has mean m_{t-1} (which is actually a linear function of x_{t-}) and variance V_{t-1}, which we may express as

$$S_{t-1} = m_{t-1} + P'_{t-1} b_{t-1} \tag{9}$$

where P_{t-1} is the right factor of V_{t-1}, such that $P'_{t-1} P_{t-1} = V_{t-1}$, and b_{t-1} has identity variance matrix. The state transition equation (2) we also write with the error term expressed from (4) as

$$S_t = T\, S_{t-1} + R'_E\, e_t, \tag{10}$$

where e_t has identity variance matrix. Combining these last two equations gives $P(S_t|x_{t-})$ expressed as

$$S_t = T\, m_{t-1} + \left(\begin{array}{c|c} T\, P'_{t-1} & R'_E \end{array} \right) \left(\frac{b_{t-1}}{e_t} \right) = n_t + A' \left(\frac{b_{t-1}}{e_t} \right) \tag{11}$$

where $n_t = T\, m_{t-1}$ and

$$A = \left(\frac{P_{t-1}\, T'}{R_E} \right).$$

Now applying the QR decomposition to A gives

$$A = Q\, U_t$$

and allows us to re-express (11) as

$$S_t = n_t + U'_t\, a_t \tag{12}$$

where U_t is square and of the same size as the state vector S_t, and a_t has mean zero and identity variance matrix. We can confirm that the variance of S_t (conditional upon x_{t-}) in this equation is given according to the simple rule for adding uncorrelated variables as

$$W_t = T\, V_{t-1} T' + V_E = T P'_{t-1} P_{t-1} T' + R'_E R_E = A'A = U'_t\, U_t. \tag{13}$$

The quantities n_t and U_t defining $P(S_t|x_{t-})$ in (12) are now used in the second step of the recursion.

Our implementation of this second step is to extend the predicting set of variables by each element of x_t in turn, updating the predicted mean and variance of S_t as each one is introduced, up to the full set of elements. This is quite efficient and has one useful property that the successive prediction errors of the elements of $x_1, \ldots, x_n$, that arise in this procedure, constitute an orthogonalization of these variables that can be used to compute the likelihood of the model.

For convenience of notation, to avoid subscripting, we will take (1) as representing a scalar observation of a particular element. The first term in (8) is simply expressed as the observation equation (1), i.e. $x_t = H\, S_t$, and the second we have just derived as (12) from the first step. Combining these, by substituting from the second into the first, expresses $P(S_t, x_t|x_{t-})$ in (8) in partitioned form as

$$\left(\frac{x_t}{S_t} \right) = \left(\frac{p_t}{n_t} \right) + \left(\frac{f'_t}{U'_t} \right) a_t = \left(\frac{p_t}{n_t} \right) + M'_t a_t \tag{14}$$

where $p_t = H\, n_t$ is the expected or predicted value of x_t given x_{t-1} and $f'_t\, a_t = H\, U' t\, a_t$ is the error in that prediction, f_t being a column vector of length d, and M_t being of size $d \times (d+1)$.

To complete the update we construct a simple orthonormal matrix N_t of size d which transforms f_t to a vector of the form $(c, 0, \ldots, 0)'$ in which all elements except the first are zero. We use this to transform the error term in (14) into an equivalent representation in terms of uncorrelated variables, of the form

$$\left(\frac{x_t}{S_t} \right) = \left(\frac{p_t}{n_t} \right) + (M'_t N'_t)\, (N_t\, a_t) = \left(\frac{p_t}{n_t} \right) + \left(\begin{array}{c|ccc} c_t & 0 & \cdots & 0 \\ \hline k_t & & P'_t & \end{array} \right) \left(\frac{\beta_t}{b_t} \right). \tag{15}$$

Because x_t is observed we can evaluate its scalar (normalized) prediction error

$$\beta_t = (x_t - p_t)/c_t$$

and then substitute back for β_t to obtain $P(S_t|x_t, x_{t-})$ in the form

$$S_t = (n_t + \beta_t\, k_t) + P_t'\, b_t = m_t + P_t\,' b_t. \tag{16}$$

This completes the recursion for $m_t = n_t + \beta_t\, k_t$ and P_t because b_t has identity variance matrix. We note however that b_t is of length $(d-1)$ and P_t is of size $(d-1) \times d$ reflecting the singular dependence between the elements of the prediction of S_t.

The simple orthonormal matrix N_t of size d which transforms f_t to a vector of the form $(c, 0, \ldots, 0)'$ is a Householder reflection of the form

$$N_t = I - 2\, v_t\, v_t',$$

where v_t is a vector determined by f_t. It is efficiently applied to the matrix M_t' in (15) as

$$M_t' N_t' = M_t' - 2\,(M_t' v_t)\, v_t'.$$

When x_t is not a scalar, the second step of filtering is applied in turn to each element $x_{i,t} = h_i\, S_t$, where h_i is row i of H, before the next application of the first step.

To start the recursions at time $t = 1$ the value of $m_0 = 0$ is used, but P_0 needs to be set non-zero in general. One method for doing this is to run the recursions for P_t from a time $t = -N$ in the distant past with $P_{-N} = 0$, omitting the second step of the recursion because no observations are present. Thus P_t is derived for $t = 1 - N, \ldots, 0$ from the QR step

$$\begin{pmatrix} P_{t-1}\, T' \\ \hline R_E \end{pmatrix} = Q P_t.$$

There is method widely used when powering matrices by successive squaring, that can be extended to speed up this recursion dramatically. Thus only K applications of QR factorization are needed for $N = 2^K$. This is implemented in the code. If the transition matrix T has all its eigenvalues less than 1 in magnitude, P_t will be close to the right square root of the stationary variance of S_t. Otherwise it will generally take a large value, representing uncertainty in the value of the state before observations are made.

4 The recursive step of smoothing

The recursive steps of smoothing are applied in a reverse time sequence. Let x_{t+} be the set of all future variables $\{x_{t+1}, x_{t+2}, \ldots, x_n\}$ at time t. The whole set of variables we will write as $x = \{x_{t-}, x_t, x_{t+}\}$. We wish to derive a recursion for $P(S_t|x)$ from $P(S_{t+1}|x)$. The CIG in Figure 1(b) allows us to do this by exploiting its property:

$$P(S_t|S_{t+1}, x) = P(S_t|S_{t+1}, x_{t-}, x_t, x_{t+}) = P(S_t|S_{t+1}, x_{t-}, x_t). \tag{17}$$

Using this gives the simplification

$$\begin{aligned} P(S_t|x) = \int_{S_{t+1}} P(S_t, S_{t+1}|x) &= \int_{S_{t+1}} P(S_t|S_{t+1}, x)P(S_{t+1}|x) \\ &= \int_{S_{t+1}} P(S_t|S_{t+1}, x_{t-}, x_t)P(S_{t+1}|x). \end{aligned} \tag{18}$$

So to implement our recursion we need

$$\begin{aligned} P(S_t|S_{t+1}, x_{t-}, x_t) \propto P(S_t, S_{t+1}|x_{t-}, x_t) &= P(S_{t+1}|S_t, x_{t-}, x_t)P(S_t|x_{t-}, x_t) \\ &= P(S_{t+1}|S_t)P(S_t|x_{t-}, x_t), \end{aligned} \tag{19}$$

where we have again exploited a simplification of the CIG.

In a similar manner to constructing (14) we use $P(S_{t+1}|S_t)$ expressed as $S_{t+1} = TS_t + R'_E\, e_{t+1}$ and, from (16), $P(S_t|x_{t-}, x_t)$ expressed as $S_t = m_t + P'_t\, b_t$ where e_{t+1} and b_t are uncorrelated and both have identity variance. Together they supply $P(S_t, S_{t+1}|x_{t-}, x_t)$ in the form

$$\left(\frac{S_{t+1}}{S_t}\right) = \left(\frac{n_{t+1}}{m_t}\right) + \left(\begin{array}{c|c} T\,P'_t & R'_E \\ \hline P'_t & 0 \end{array}\right)\left(\frac{b_t}{e_{t+1}}\right). \tag{20}$$

where $n_{t+1} = T\,m_t$ The derivation, from this, of an expression for $P(S_t|S_{t+1}, x_{t-}, x_t)$ is, however, complicated by the possibility that the variance matrix of S_{t+1}, as implied by (20), may be singular. To allow for this, we assume that the elements of S_{t+1} have been ordered and partitioned as in

$$S_{t+1} = \left(\frac{S^{(1)}_{t+1}}{S^{(2)}_{t+1}}\right) \tag{21}$$

where $S^{(1)}_{t+1}$ are non-singular with positive definite variance and each element of $S^{(2)}_{t+1}$ is linearly dependent on elements of $S^{(1)}_{t+1}$. We may then apply QR decomposition to the transpose of the partitioned matrix in (20) to give

$$\left(\begin{array}{c|c} P_t\,T' & P_t \\ \hline R_E & 0 \end{array}\right) = Q\left(\begin{array}{c|c} \left[\begin{array}{c|c} A & B \\ \hline 0 & 0 \end{array}\right] & \left[\begin{array}{c} C \\ \hline D \end{array}\right] \\ \hline 0 & E \end{array}\right) \tag{22}$$

with A and E being upper triangular. All the sub-matrices on the right in (22) depend on t, but for simplicity of notation this is not indicated. The ordering of the states in (21) is achieved by a standard modification of the QR decomposition applied to the first column of the partitioned matrix in (22). This produces a triangular factor with columns ordered for reducing magnitude of diagonal elements. Any dependent columns correspond to terminating zeros on the diagonal and, in fact, identically zero rows of the triangular factor.

Substitution of (22) into (20) gives equations for the states as:

$$\begin{array}{rcccccccc} S^{(1)}_{t+1} &=& n^{(1)}_{t+1} &+& A'r^{(1)}_{t+1} & & & & \\ S^{(2)}_{t+1} &=& n^{(2)}_{t+1} &+& B'r^{(1)}_{t+1} & & & & \\ S_t &=& m_t &+& C'r^{(1)}_{t+1} &+& D'r^{(2)}_{t+1} &+& E'r^{(3)}_{t+1} \end{array}, \tag{23}$$

where we have partitioned n_{t+1} in the same manner as S_{t+1} and the terms $r_{t+1}^{(1)}$ for $i = 1, 2, 3$ are conformable partitions of an error vector r_{t+1} with identity variance matrix. From the first and third of these equations we obtain the required representation of $P(S_t|S_{t+1})$ as the regression

$$S_t = m_t + J_t\,(S_{t+1} - n_{t+1}) + F_t'\,g_t \tag{24}$$

where

$$J_t = \left(\; C'\,(A')^{-1} \;\middle|\; 0 \;\right), \tag{25}$$

F_t is derived by the QR decomposition

$$\left(\frac{D}{E}\right) = Q\,F_t \tag{26}$$

and g_t has identity variance matrix.

For the final step of the smoothing recursion we take $P(S_{t+1}|x)$ expressed using the square root form of the error as

$$S_{t+1} = u_{t+1} + G_{t+1}'\,z_{t+1} \tag{27}$$

which, following (18), we simply substitute into (24), to give the expression for $P(S_t|x)$ as

$$\begin{aligned} S_t &= [m_t + J_t\,(u_{t+1} - n_{t+1})] + \left[J_t\,G_{t+1}'\,z_{t+1} + F_t'\,g_t\right] \\ &= u_t + G_t'\,z_t \end{aligned} \tag{28}$$

where QR decomposition has been applied to give

$$\left(\frac{G_{t+1}\,J_t'}{F_t}\right) = Q\,G_t. \tag{29}$$

References

G. Tunnicliffe Wilson, M. Reale, and J. Haywood. *Models for dependent time series*. New York, CRC Press, 2015.