

## 1 The Kalman Filter

We assume that we have a model that concerns a series of vectors  $\alpha_t$ , which are called “state vectors”. These variables are supposed to describe the current state of the system in question. These state variables will typically not be observed and the other main ingredient is therefore the observed variables  $y_t$ . The first step is to write the model in **state space form** which is in the form of a linear system which consists of 2 sets of linear equations. The first set of equations describes the evolution of the system and is called the “Transition Equation”:

$$\alpha_t = K\alpha_{t-1} + R\eta_t ,$$

where  $K$  and  $R$  are matrices of constants and  $\eta$  is  $N(0, Q)$ . The second set of equations describes the relation between the state of the system and the observations and is called the “Measurement Equation”:

$$y_t = Z\alpha_t + \xi_t ,$$

where  $\xi_t$  is  $N(0, H)$  and  $E(\xi_t\eta_{t-j}) = 0$  for all  $t$  and  $j$ .

It turns out that a lot of models can be put in the state space form with a little imagination. The main restriction is of course on the linearity of the model. The state-space model as it is defined here is not the most general possible – it is in principle easy to allow for non-stationary coefficient matrices, see for example Harvey(1989). There is also extension of Kalman filter methods to non-linear models. These are known as extended Kalman filters and they are also treated in Harvey (1989) – be aware, however, that extended Kalman filters usually can not be used to evaluate likelihood functions exactly; but only gives an approximation.

As an example, let us look at a model where the economy consists of two sectors producing a homogeneous product, where we only observe the aggregate output subject to measurement error. Assume that the output of the 2 sectors follow a VAR(1) model. Then the state-space system becomes as follows. Transition equation:

$$\begin{pmatrix} \alpha_t^1 \\ \alpha_t^2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \alpha_{t-1}^1 \\ \alpha_{t-1}^2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_t^1 \\ \eta_t^2 \end{pmatrix} ,$$

and measurement equation:

$$y_t = (1 \ 1) \begin{pmatrix} \alpha_t^1 \\ \alpha_t^2 \end{pmatrix} + \xi_t ,$$

ARMA processes can also be written in State-Space form and we will use the Kalman filter to estimate the likelihood function for ARMA processes. In the case of a general ARMA process, one can use several representations; but the most compact (and useful) one is the following. Assume that we have given the scalar ARMA process (where I leave out the mean for simplicity):

$$x_t = a_1 X_{t-1} + \dots + a_k X_{t-m} + u_t + b_1 u_{t-1} + \dots + b_l u_{t-l} ,$$

where  $m = \max\{k, l + 1\}$ . This process can be represented in the following state space form:  
Transition equation

$$\alpha_t = \begin{pmatrix} a_1 & 1 & 0 & \dots & 0 \\ a_2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m-1} & 0 & 0 & \dots & 1 \\ a_m & 0 & 0 & \dots & 0 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{pmatrix} u_t ,$$

and measurement equation

$$x_t = (1, 0, \dots, 0) \alpha_t .$$

*Example.* The MA(1) model has the state-space representation

$$\alpha_t = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} 1 \\ b_1 \end{pmatrix} u_t .$$

If  $\alpha_t = (\alpha_{1t}, \alpha_{2t})'$ , then  $\alpha_{2t} = b_1 u_t$  and  $\alpha_{1t} = \alpha_{2,t-1} + u_t = u_t + b_1 u_{t-1}$ .

It is an exercise to show that the general ARMA-case does indeed have the state-space representation given above.

The Kalman filter is very useful when you want to calculate the likelihood function. You will typically have a general maximization algorithm at your disposal (e.g. the OPTMUM algorithm in GAUSS). Such an algorithm takes as input a subroutine that evaluates the value of the likelihood function for a given set of parameters. This is where you will use the Kalman filter algorithm.

For a start, look at the general likelihood function:

$$f(y_T, \dots, y_1; \theta) ,$$

where  $\theta$  is a vector of parameters. One can always factor such a likelihood function (independent of which model or distribution that generates the variables), as

$$f(y_T, \dots, y_1; \theta) = f(y_T | y_{T-1}, \dots, y_1; \theta) f(y_{T-1}, \dots, y_1; \theta) .$$

Iterating this formula we find

$$f(y_T, \dots, y_1; \theta) = \prod_{t=p+1}^T f(y_t | y_{t-1}, \dots, y_1; \theta) f(y_p, \dots, y_1; \theta) ,$$

From which we find the log-likelihood function in *recursive form* as

$$L = \ln f = \sum_{t=p+1}^T \ln f(y_t | y_{t-1}, \dots, y_1; \theta) + \ln f(y_p, \dots, y_1; \theta) .$$

In the case of the normal likelihood function this becomes

$$L = \sum_{t=p+1}^T -\frac{1}{2} \ln |F_t| - \frac{1}{2} \nu_t' F_t^{-1} \nu_t + \text{constant} + \ln f(y_p, \dots, y_1; \theta) ,$$

where

$$\nu_t = y_t - E(y_t | y_{t-1}, \dots, y_1) \text{ and } F_t = E[\nu_t \nu_t' | y_{t-1}, \dots, y_1] .$$

Now if we knew  $F_t$  and  $\nu_t$  we would have a quite convenient way of evaluating the likelihood function. This is what the Kalman filter equations below are designed to do.

At this stage note the following aside. The likelihood equations in recursive form allows you to evaluate the “impact” of a new observation arriving, in the sense that it immediately shows the conditional likelihood. In engineering it is often important to be able to *update* a parameter estimate instantly when a new observation occurs - and hopefully without having to reestimate using all the data. The Kalman Filter does exactly that and it is therefore used extensively by engineers. More surprising is the fact that it at the same time is so convenient to use that it is also a good choice to use for the purpose of a single estimation on a given data set.

The ingredients of the Kalman filter (besides the state-space representation) consist of *predicting equations* and *updating equations*.

For any vector  $x_t$  define  $x_{t|t-1} = E(x_t | y_{t-1}, \dots, y_1)$ , where  $y_j$  are the *observed* variables. This definition gives the best guess of  $x_t$  based on all the information available at time  $t-1$ ,  $x_{t|t-1}$  is the prediction of  $x_t$  at  $t-1$ . As you may guess, the Kalman filter evolves around predicting and updating the prediction of the state vector. Also define  $P_{t|t-1} = E\{(\alpha_t - \alpha_{t|t-1})(\alpha_t - \alpha_{t|t-1})'\} - P_{t|t-1}$  is the conditional variance of the “prediction error”  $\alpha_t - \alpha_{t|t-1}$ .

We will first describe the Kalman filter and then derive it.

The prediction equations take the form

$$\begin{aligned}\alpha_{t|t-1} &= K\alpha_{t-1|t-1} \\ y_{t|t-1} &= Z\alpha_{t|t-1} \\ P_{t|t-1} &= KP_{t-1|t-1}K' + RQR' .\end{aligned}$$

If we define

$$\nu_t = y_t - y_{t|t-1} ,$$

the variance matrix

$$F_t = E\{\nu_t\nu_t'\} = E\{(y_t - y_{t|t-1})(y_t - y_{t|t-1})'\} = ZP_{t|t-1}Z' + H .$$

This follows since

$$y_t - y_{t|t-1} = Z(\alpha_t - \alpha_{t|t-1}) + \xi_t .$$

To finish the Kalman filter we need the updating equations:

$$\begin{aligned}\alpha_{t|t} &= \alpha_{t|t-1} + P_{t|t-1}Z'F_t^{-1}\nu_t \\ P_{t|t} &= P_{t|t-1} - P_{t|t-1}Z'F_t^{-1}ZP_{t|t-1} .\end{aligned}$$

The interpretation of the updating equations is that  $\nu_t$  contains the new information (from  $y_t$ ) and we update our estimate of  $\alpha_t$  based on  $y_1, \dots, y_{t-1}$  (i.e.  $\alpha_{t|t-1}$ ) to a new estimate that is based on  $y_1, \dots, y_{t-1}$  and  $y_t$  (the new estimate is  $\alpha_{t|t}$ ), and we calculate the conditional variance  $P_{t|t}$  of  $\alpha_{t|t}$ . The term

$$P_{t|t-1}Z'F_t^{-1}\nu_t ,$$

is called the *Kalman gain*. Any new information enters the system through the Kalman gain.

The Kalman filter can be derived from the rules of the Normal distribution:

We can write

$$\begin{aligned}\nu_t &= Z(\alpha_t - \alpha_{t|t-1}) + \xi_t \\ \alpha_t &= \alpha_{t|t-1} + (\alpha_t - \alpha_{t|t-1}) ,\end{aligned}$$

From our definitions we have

$$\begin{pmatrix} \nu_t \\ \alpha_t \end{pmatrix} \Big|_{y_{t-1}, \dots, y_1} = N \left( \begin{pmatrix} 0 \\ \alpha_{t|t-1} \end{pmatrix}, \begin{bmatrix} F_t & ZP_{t|t-1} \\ P_{t|t-1}Z' & P_{t|t-1} \end{bmatrix} \right)$$

Recall the following rule for the conditional Normal distribution (or see e.g. Lütkepohl (1991), pp. 480-81). If

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathbf{N} \left( \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right),$$

then the conditional distribution of  $x_1$  given  $x_2$  is

$$N(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), [\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}]).$$

Using this rule on the conditional distribution of  $(\nu_t, \alpha_t)$  we find

$$\mathcal{L}(\alpha_t | y_t, \dots, y_1) = \mathcal{L}(\alpha_t | \nu_t, y_{t-1}, \dots, y_1) = N(\alpha_{t|t-1} + P_{t|t-1}Z'F_t^{-1}\nu_t, P_{t|t-1} - P_{t|t-1}Z'F_t^{-1}ZP_{t|t-1}),$$

which is the updating equation.

One problem is how to initialize the filter. It is natural to choose  $\alpha_{t|t-1} = 0$  since this is the unconditional mean of  $\alpha_t$ . It is often also natural to choose the stationary value of the variance as the initial value for the variance, even though this is only an option in the stable case). Other choices are possible, for example you may want to condition on initial values as discussed earlier; but in that case the special cases has to be considered one by one. One can, however, find the stationary variance by the following method.

Combining the updating and the prediction equations we find

$$P_{t+1|t} = KP_{t|t-1}K' - KP_{t|t-1}Z'(ZP_{t|t-1}Z' + H)^{-1}ZP_{t|t-1}K' + RQR',$$

which is known as the Riccati equation. If the model is stable  $P_{t|t-1}$  will converge to the solution  $\bar{P}$  of the *algebraic Ricatti equation*

$$\bar{P} = K\bar{P}K' - K\bar{P}Z'(Z\bar{P}Z' + H)^{-1}Z\bar{P}K' + RQR'.$$

In order to apply the Kalman filter one has to choose a set of starting values. The most natural choice for a stable system is the unconditional mean and variance. Since

$$\alpha_t = K\alpha_{t-1} + R\eta_t,$$

has the form of an AR(1) model, we will then choose  $\alpha_{1|0} = 0$  and  $P_0$  such that

$$\text{vec}(P_0) = (I - K \otimes K)^{-1} \text{vec}(RQR').$$

If you want, you can choose other initial conditions, for example chosen from a Bayesian prior, or if you want to condition on initial values. In the non-stationary case it is obviously not

possible to choose the initial distribution from the stationary distribution.

*Example:* The state-space representation for an AR(2) model is

$$\begin{pmatrix} x_t \\ a_2 x_{t-1} \end{pmatrix} = \begin{pmatrix} a_1 & 1 \\ a_2 & 0 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ a_2 x_{t-2} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u_t,$$

so here conditioning on initial observations just corresponds to an initial variance of zero; but in the ARMA case one has to be a bit more sophisticated unless one want to condition on initial values of the innovation terms to be zero. This is not always advisable as discussed previously.

Now in order to use the Kalman filter you will need a general optimization routine as OPTMUM in GAUSS. Such a routine need a subroutine that returns the criterion function and the optimization algorithm will then use some specified algorithm to minimize (or maximize) the value of the criterion function. Let me sketch how a subroutine using the Kalman filter for evaluating a Normal likelihood function would look. Assume the subroutine evaluating the likelihood function is called CRITFUNC. If we evaluate the value of the likelihood function for a scalar ARMA model it will have the parameters  $a_1, \dots, a_k, b_1, \dots, b_l, \sigma$  of the ARMA model as arguments. The structure of a GAUSS subroutine would be something like this:

```
CRITFUNC(a1, ...ak, b1, ..., bl, sigma);
@ First create the matrices that is used in state-space form @
K[1, 1] = a1;
K[1, 2] = a2;
etc.
@ Initialize @
L = 0 @ L is the value of the likelihood function @
t = 0
@Initialize the Kalman Filter.@
P0 = ..
etc.
@ Loop @
DO UNTIL t == (T-1)

Prediction Equations for time t

Evaluate the condition likelihood = L(yt|yt-1, ..., y1)
L = L + L(yt|yt-1, ..., y1)

Updating equations for t ;
```

```
ENDO ;  
@return the value of the likelihood function for all points@  
RETURN(L);  
END OF CRITFUN
```