STATIC AND DYNAMIC RECURSIVE LEAST SQUARES

3rd February 2006

1 Problem #1: additional information

Problem. At step k we want to solve by least squares

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix} x_k \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}, \qquad \mathcal{A}_k := \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix}, \qquad \mathbf{b}_k := \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}$$

with weight matrix

$$\mathcal{W}_k := \begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_k \end{bmatrix}$$

meaning that measurements at different times (steps) are independent.

Sizes.

- Matrices A_j are $m_j \times n$. We assume that they have maximum column rank.
- Matrices W_j are symmetric and positive definite.
- **Simple case.** $m_j = m$ for all j, meaning that the number of observations is constant.
- **Magnitudes.** We assume that n is small, whereas m_j can be relatively large.

The aim. To approximate $\mathcal{A}_k x_k \approx \mathbf{b}_k$ we solve

$$(\mathcal{A}_k^\top \mathcal{W}_k \mathcal{A}_k)^{-1} \mathcal{A}_k^\top \mathcal{W}_k \mathbf{b}_k.$$

The idea is to use x_{k-1} to calculate x_k in a shorter time.

Notation. We write

$$P_k^{-1} := \sum_{j=1}^{k} A_j^\top W_j A_j = \mathcal{A}_k^\top \mathcal{W}_k \mathcal{A}_k.$$

Theorem (A recursion) *Beginning with*

$$x_1 = P_1 A_1^\top W_1 b_1, \qquad P_1 = (A_1^\top W_1 A_1)^{-1}$$

the solution is given by the recurrent calculations $j = 1, \ldots, k$

$$x_j = x_{j-1} + \underbrace{P_j A_j^\top W_j}_{=: K_j}(b_j - A_j x_{j-1}), \qquad P_j^{-1} = P_{j-1}^{-1} + A_j^\top W_j A_j.$$

Proof. It is a straightforward verification. □

Remark. The vector $b_j - A_j x_{j-1}$ is the residual of the observations at level j by taking into account the values of the parameters at level $j-1$.

A practical algorithm.

$P^{-1} := A_1^\top W_1 A_1$
$x :=$**solve**$(P^{-1}, A_1^\top W_1 b_1)$
for $j = 1 : k$
  $P^{-1} := P^{-1} + A_j^\top W_j A_j$
  $r := b_j - A_j x$ % residual of recent calculations with new data
  $\delta :=$**solve**$(P^{-1}, A_j^\top W_j r)$
  $x := x + \delta$
end for

Computational features. Explicit assembling of P^{-1} could be avoided if iterative methods were to be used.

2 Problem # 2: dynamic least squares

The problem. Least squares solution to

$$\begin{bmatrix} A_0 & & & & \\ -F_0 & I & & & \\ & A_1 & & & \\ & -F_1 & I & & \\ & & \ddots & & \\ & & -F_{k-1} & I \\ & & & A_k \end{bmatrix} \begin{bmatrix} x_{0|k} \\ x_{1|k} \\ \vdots \\ x_{k-1|k} \\ x_{k|k} \end{bmatrix} \approx \begin{bmatrix} b_0 \\ 0 \\ b_1 \\ 0 \\ \vdots \\ 0 \\ b_k \end{bmatrix}$$

with weight matrix

$$\mathcal{W}_k = \begin{bmatrix} W_0 & & & & & \\ & \widetilde{W}_1 & & & & \\ & & W_1 & & & \\ & & & \ddots & & \\ & & & & \widetilde{W}_k & \\ & & & & & W_k \end{bmatrix}$$

Interpretation. Observations in different times (for different magnitudes) are contained in the equations

$$A_j x_j \approx b_j$$

whereas

$$x_{j+1} \approx F_j x_j$$

is an approximate dynamic model, relating the sets of parameters.

Simplified notations. For matrices we write

$$\mathcal{A}_k := \begin{bmatrix} A_0 & & & & \\ -F_0 & I & & & \\ & A_1 & & & \\ & -F_1 & I & & \\ & & \ddots & & \\ & & -F_{k-1} & I \\ & & & A_k \end{bmatrix}, \qquad \mathcal{S}_k := \begin{bmatrix} A_0 & & & \\ -F_0 & I & & \\ & A_1 & & \\ & -F_1 & I & \\ & & \ddots & \\ & & -F_{k-1} & I \end{bmatrix}$$

Names.

- smoothed values (at earlier states): $x_{j|k}$ for $j < k$;
- filtered value (at the current state): $x_{k|k}$;
- predicted value (next state in the future, not yet observed): $x_{k+1|k} := F_k x_{k|k}$.

Theorem (Kalman filter recurrence) *Starting with*

$$x_{0|0} := (A_0^\top W_0 A_0)^{-1} A_0^\top W_0 b_0, \qquad P_{0|0} := (A_0^\top W_0 A_0)^{-1},$$

the calculations for $x_{k|k}$ at different values of k can be carried out by doing:

$$x_{k|k-1} := F_{k-1} x_{k-1|k-1}, \qquad x_{k|k} := x_{k|k-1} + K_k (b_k - A_k x_{k|k-1})$$

where

$$\begin{aligned} P_{k|k-1} &:= F_{k-1} P_{k-1|k-1} F_{k-1}^\top + \widetilde{W}_k^{-1} \\ K_k &:= P_{k|k-1} A_k^\top (A_k P_{k|k-1} A_k^\top + W_k^{-1})^{-1} \\ P_{k|k} &:= (I - K_k A_k) P_{k|k-1} \end{aligned}$$

Proof. See Section 4. □

Names.

- For the two steps of the method we write:
  - prediction $x_{k|k-1} := F_{k-1}x_{k-1|k-1}$
  - correction $x_{k|k} := x_{k|k-1} + K_k(b_k - A_k x_{k|k-1})$
- For the matrices intervening in computations
  - predicted covariance $P_{k|k-1}$
  - gain matrix K_k
  - corrected covariance $P_{k|k}$

Sizes in the typical simple case.

- The number of parameters in the dynamical system is fixed, i.e., $x_{j|k} \in \mathbb{R}^m$ for all j, k
- The number of observations in each times step is fixed, i.e., $b_j \in \mathbb{R}^n$ for all j.

$$m = \#\{\text{parameters}\}, \qquad n = \#\{\text{observations}\}.$$

- Hence the sizes of the matrices are as follows:
  - W_j is $m \times m$
  - $\widehat{W}_j$ is $n \times n$
  - F_j is $m \times m$
  - A_j is $n \times m$
  - $P_{k|k}$ and $P_{k|k-1}$ are $m \times m$
  - K_k is $m \times n$ (acts on observations returning parameters).

Computational features. When n is large and m is small

- We have to compute $C_k = A_k P_{k|k-1} A_k^\top + W_k^{-1}$, which is $n \times n$. Typically we have W_k^{-1} instead of W_k.
- To compute $K_k(b_k - A_k x_{k|k-1}) = P_{k|k-1} A_k^\top C_k^{-1}(b_k - A_k x_{k|k-1})$ we solve an $n \times n$ system.
- To compute $K_k A_k = P_{k|k-1} A_k^\top C_k^{-1} A_k$ we solve m systems $n \times n$. All these calculations can be done in parallel.

Steady state. We remark that the steady–state case, where the dynamics reduces to

$$x_{j+1} \approx x_j, \qquad \forall j$$

that is,

$$F_j = I, \qquad \forall j$$

is not equivalent to the static case (Problem #1).

3 Normal equations of the Kalman filter

A first observation. In the normal equations

$$\mathcal{A}_k^\top \mathcal{W}_k \mathcal{A}_k \mathbf{x}_k = \mathcal{A}_k^\top \mathcal{W}_k \mathbf{b}_k$$

we can write

$$\mathcal{A}_k^\top \mathcal{W}_k \mathcal{A}_k = \begin{bmatrix} D_0 & U_0 & & & \\ U_0^\top & D_1 & U_1 & & \\ & U_1^\top & \ddots & \ddots & \\ & & \ddots & D_{k-1} & U_{k-1} \\ & & & U_{k-1}^\top & D_k^\bullet \end{bmatrix}, \qquad \mathcal{A}_k^\top \mathcal{W}_k \mathbf{b}_k = \begin{bmatrix} A_0^\top W_0 b_0 \\ A_1^\top W_1 b_1 \\ \vdots \\ A_k^\top W_k b_k \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_k \end{bmatrix}$$

where (taking $\widetilde{W}_0 = 0$)

$$\begin{aligned} D_j &:= \widetilde{W}_j + A_j^\top W_j A_j + F_j^\top \widetilde{W}_{j+1} F_j \\ U_j &:= -F_j^\top \widetilde{W}_{j+1} \\ D_k^\bullet &:= \widetilde{W}_k + A_k^\top W_k A_k. \end{aligned}$$

The symbol $\bullet$ marks the only element of the matriz which depends on the size (k). Notice also that for the next time step

$$D_k = D_k^\bullet + F_k^\top \widetilde{W}_{k+1} F_k$$

Another look at the normal equations. The system

$$\begin{bmatrix} D_0 & U_0 & & & \\ U_0^\top & D_1 & U_1 & & \\ & U_1^\top & \ddots & \ddots & \\ & & \ddots & D_{k-1} & U_{k-1} \\ & & & U_{k-1}^\top & D_k^\bullet \end{bmatrix} \begin{bmatrix} x_0^\bullet \\ x_1^\bullet \\ \vdots \\ x_{k-1}^\bullet \\ x_k^\bullet \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \\ c_k \end{bmatrix}$$

is symmetric block tridiagonal.

Reduced equations. Before considering the inclusions of observations at time k ($A_k x_k \approx b_k$) we can consider the reduced system

$$\begin{bmatrix} A_0 & & & & \\ -F_0 & I & & & \\ & A_1 & & & \\ & -F_1 & I & & \\ & & \ddots & & \\ & & & -F_{k-1} & I \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \\ y_k \end{bmatrix} \approx \begin{bmatrix} b_0 \\ 0 \\ \vdots \\ b_{k-1} \\ 0 \end{bmatrix}$$

Theorem *The vector*

$$(x_{0|k-1}, x_{1|k-1}, \ldots, x_{k-1|k-1}, \underbrace{F_{k-1} x_{k-1|k-1}}_{x_{k|k-1}})^\top$$

is the weighted least square solution to the reduced system.

Proof. Denote $\mathcal{S}_k$ as above, and

$$\widetilde{\mathcal{W}}_k = \begin{bmatrix} W_0 & & & & \\ & \widetilde{W}_1 & & & \\ & & W_1 & & \\ & & & \ddots & \\ & & & & \widetilde{W}_k \end{bmatrix}$$

to the weight matrix. The matrix for the normal equations is

$$\begin{aligned} \mathcal{S}_k^\top \widetilde{\mathcal{W}}_k \mathcal{S}_k &= \begin{bmatrix} D_0 & U_0 & & & \\ U_0^\top & D_1 & U_1 & & \\ & U_1^\top & \ddots & \ddots & \\ & & \ddots & D_{k-1} & U_{k-1} \\ & & & U_{k-1}^\top & \widetilde{W}_k \end{bmatrix} = \\ &= \begin{bmatrix} \mathcal{A}_{k-1}^\top \mathcal{W}_{k-1} \mathcal{A}_{k-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \\ 0 & F_{k-1}^\top \widetilde{W}_k F_{k-1} & U_{k-1} \\ & U_{k-1}^\top & \widetilde{W}_k \end{bmatrix} \end{aligned}$$

and the right–hand side is

$$(c_0, \ldots, c_{k-1}, 0)^\top.$$

It is simple to prove that the proposed solution satisfies the equations. □

A direct block method. If we only want to solve the last unknown $x_k^\bullet = x_{k|k}$, we can apply an elimination technique to reach the upper block triangular form

$$\begin{bmatrix} E_0 & U_0 & & & \\ & E_1 & U_1 & & \\ & & \ddots & \ddots & \\ & & & E_{k-1} & U_{k-1} \\ & & & & E_k^\bullet \end{bmatrix} \begin{bmatrix} x_0^\bullet \\ x_1^\bullet \\ \vdots \\ x_{k-1}^\bullet \\ x_k^\bullet \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \\ d_k \end{bmatrix}$$

by using the well–known tridiagonal algorithm

$E_0 := D_0$
$d_0 := c_0$
for $j = 1 : k-1$
  $E_j := D_j - U_{j-1}^\top E_{j-1}^{-1} U_{j-1}$
  $d_j := c_j - U_{j-1}^\top E_{j-1}^{-1} d_{j-1}$
end for
$E_k^\bullet := D_k^\bullet - U_{k-1}^\top E_{k-1}^{-1} U_{k-1}$
$d_k := c_k - U_{k-1}^\top E_{k-1}^{-1} d_{k-1}$
$x_k :=$**solve**$(E_k^\bullet, d_k)$

The same elimination process applied to the reduced equations yields

$$\begin{bmatrix} E_0 & U_0 & & & \\ & E_1 & U_1 & & \\ & & \ddots & \ddots & \\ & & & E_{k-1} & U_{k-1} \\ & & & & H_k^\bullet \end{bmatrix} \begin{bmatrix} y_0^\bullet \\ y_1^\bullet \\ \vdots \\ y_{k-1}^\bullet \\ y_k^\bullet \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \\ e_k^\bullet \end{bmatrix}$$

where

$$\begin{aligned} H_k^\bullet &:= \widetilde{W}_k - U_{k-1}^\top E_{k-1}^{-1} U_{k-1} = E_k^\bullet - A_k^\top W_k A_k \\ e_k^\bullet &:= -U_{k-1}^\top E_{k-1}^{-1} d_{k-1} = d_k - A_k^\top W_k b_k = d_k - c_k. \end{aligned}$$

The next step. The equality $D_k = D_k^\bullet + F_k^\top \widetilde{W}_{k+1} F_k$ implies, that the next step only requires the following calculations

$$\begin{aligned} E_k &:= E_k^\bullet + F_k^\top \widetilde{W}_{k+1} F_k \\ D_{k+1}^\bullet &:= \widetilde{W}_{k+1} + A_{k+1}^\top W_{k+1} A_{k+1} \\ U_k &:= -F_k^\top \widetilde{W}_{k+1} \\ E_{k+1}^\bullet &:= D_{k+1}^\bullet - U_k^\top E_k^{-1} U_k \\ d_{k+1} &:= c_{k+1} - U_k^\top E_k^{-1} d_k \end{aligned}$$

The method as a whole. Notice that we can write all the steps in the preceding recursive way. Notice however that this scheme does not use the solutions at previous states, unlike the true Kalman filter implementation.

$D_0^\bullet := A_0^\top W_0 A_0$
$E_0^\bullet := D_0^\bullet$
$d_0 := A_0^\top W_0 b_0$
$x_0^\bullet :=$**solve**$(E_0^\bullet, d_0)$
for $j = 1 : k$
  $E_{j-1} = E_{j-1}^\bullet + F_{j-1}^\top \widetilde{W}_j F_{j-1}$
  $D_j^\bullet := \widetilde{W}_j + A_j^\top W_j A_j$
  $U_{j-1} := -F_{j-1}^\top \widetilde{W}_j$
  $E_j^\bullet := D_j^\bullet - U_{j-1}^\top E_{j-1}^{-1} U_{j-1}$
  $d_j := A_j^\top W_j b_j - U_{j-1}^\top E_{j-1}^{-1} d_{j-1}$
  $x_j^\bullet :=$**solve**$(E_j^\bullet, d_j)$
end for

Writing the program in a more computational way, we can see the memory requirements of the algorithm.

$A := A_0$**;** $b := b_0$ % storage of observations
$W := W_0$**;** % storage of weights
$D^\bullet := A^\top W A$
$E^\bullet := D^\bullet$
$d := A^\top W b$
$x :=$**solve**$(E^\bullet, d)$
for $j = 1 : k$
  $F := F_{j-1}$**;** $\widetilde{W} := \widetilde{W}_j$ % storage of the state equation
  $U := -F^\top \widetilde{W}$
  $E = E^\bullet + F^\top \widetilde{W} F$
  $A := A_j$**;** $b := b_j$**;** $W := W_j$
  $D^\bullet := \widetilde{W} + A^\top W A$
  $E^\bullet := D^\bullet - U^\top E^{-1} U$
  $d := A^\top W b - U^\top E^{-1} d$
  $x^\bullet :=$**solve**$(E^\bullet, d)$
end for

4 A proof of the Kalman filter recursion

Lemma *Let C and P be invertible. Then the inverse of*

$$Q := P - PB^\top(BPB^\top + C^{-1})^{-1}BP$$

is

$$Q^{-1} = P^{-1} + B^\top CB.$$

Proof. It can be easily proven by direct verification. Also we can easily see that

$$\begin{bmatrix} C^{-1} + BPB^\top & -BP \\ -PB^\top & P \end{bmatrix} = \begin{bmatrix} I & 0 \\ \times & I \end{bmatrix} \begin{bmatrix} \times & \times \\ 0 & Q \end{bmatrix}$$

($\times$ is used in positions we are not interested in). Inverting both sides

$$\begin{bmatrix} C & CB \\ B^\top C & P^{-1} + B^\top CB \end{bmatrix} = \begin{bmatrix} \times & \times \\ 0 & Q^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ \times & I \end{bmatrix}$$

Comparison of blocks $(2, 2)$ gives the result. □

Theorem *For all k*

$$P_{k|k}^{-1} = E_k^\bullet, \qquad P_{k|k-1}^{-1} = H_k^\bullet.$$

Proof. The couple $(H_k^\bullet, E_k^\bullet)$ is given by the recurrence

$$\begin{aligned} H_k^\bullet &= \widetilde{W}_k - \widetilde{W}_k F_{k-1}^\top (E_{k-1}^\bullet + F_{k-1}\widetilde{W}_k F_{k-1})^{-1} F_{k-1}\widetilde{W}_k, \\ E_k^\bullet &= H_k^\bullet + A_k^\top W_k A_k \end{aligned}$$

starting at $E_0^\bullet = A_0^\top W_0 A_0$. On the other hand, the couple $(P_{k|k-1}, P_{k|k})$ satisfies the recurrence

$$\begin{aligned} P_{k|k-1} &= \widetilde{W}_k^{-1} + F_{k-1}P_{k-1|k-1}F_{k-1}^\top, \\ P_{k|k} &= P_{k|k-1} - P_{k|k-1}A_k^\top (A_k P_{k|k-1} A_k^\top + W_k^{-1})^{-1} A_k P_{k|k-1} \end{aligned}$$

starting with $P_{0|0} = (A_0^\top W_0 A_0)^{-1}$. The lemma clearly shows that both recurrences are equivalent. □

Theorem *For all k*

$$x_{k|k} = (I - K_k A_k)x_{k|k-1} + K_k b_k.$$

Proof. Notice first that by the preceding result and the form of the corresponding normal equations and their solutions

$$\begin{aligned} P_{k|k}^{-1} x_{k|k} &= d_k \\ P_{k|k-1}^{-1} x_{k|k-1} &= d_k - A_k^\top W_k b_k \end{aligned}$$

and also

$$P_{k|k}^{-1}(I - K_k A - k) = P_{k|k-1}^{-1}.$$

The result is then equivalent to showing that

$$P_{k|k} d_k = P_{k|k} d_k - P_{k|k} A_k^\top W_k b_k + K_k b_k$$

but

$$P_{k|k} A_k^\top W_k = K_k$$

as can be deduced from the definiyos of $P_{k|k}$ and K_k. □

Further reading and sources

Gilbert Strang. Introduction to Applied Mathematics. Wellesley–Cambridge.

Gilbert Strang. Linear Algebra, Geodesy and GPS. Wellesley–Cambridge.

Neil Gershhenfeld. The Nature of Mathematical Modeling. Cambridge University Press, 1999 (chapter 15).

by F.–J. Sayas

Mathematics, Ho!, May, 2001