Real-time Generalized Least Squares

Bill Menke, June 7, 2022

We imagine a generalized least squares (GLS) problem with space and time variability. The spatial part of the solution has size, M, which is repeated for $K+1$ time steps. The size of the problem grows with time, but we are most interested in the current ("last") time, $K+1$. The Gram matrix of this problem has the form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{m_1} \\ \mathbf{m_2} \end{bmatrix} = \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}$$

Here, the $M \times M$ submatrix, $\mathbf{B}$, the $M \times 1$ right hand side (r.h.s) vector, $\mathbf{a_2}$ and the $M \times 1$ solution vector, $\mathbf{m_2}$ is associated with the final, $K+1$, time step. The $MK \times MK$ submatrix, $\mathbf{A}$, the $MK \times 1$ r.h.s. vector, $\mathbf{a_1}$, and the $MK \times 1$ solution, $\mathbf{m_1}$, is associated with the previous K time steps. The submatrix, $\mathbf{C}$ is $M \times MK$.

The bordering method is used to construct the inverse of the Gram matrix. We start with the definition of the matrix inverse:

$$\begin{bmatrix} \mathbf{D} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

We then multiply out this equation and solve for the initially unknown submatrices, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$:

$$\mathbf{DA} + \mathbf{F}^T\mathbf{C} = \mathbf{I} \quad \text{so} \quad \mathbf{D} = [\mathbf{I} - \mathbf{F}^T\mathbf{C}]\mathbf{A}^{-1}$$

$$\mathbf{DC}^T + \mathbf{F}^T\mathbf{B} = \mathbf{0} \quad \text{so} \quad \mathbf{F}^T = -\mathbf{DC}^T\mathbf{B}^{-1}$$

$$\mathbf{FA} + \mathbf{EC} = \mathbf{0} \quad \textbf{so} \quad \mathbf{F} = -\mathbf{ECA}^{-1}$$

$$\mathbf{FC}^T + \mathbf{EB} = \mathbf{I} \quad \text{so} \quad -\mathbf{ECA}^{-1}\mathbf{C}^T + \mathbf{EB} = \mathbf{I} \quad \text{so} \quad \mathbf{E} = [\mathbf{B} - \mathbf{CA}^{-1}\mathbf{C}^T]^{-1}$$

$$\mathbf{F} = -\mathbf{ECA}^{-1} = -[\mathbf{B} - \mathbf{CA}^{-1}\mathbf{C}^T]^{-1}\mathbf{CA}^{-1}$$

$$\mathbf{D} = [\mathbf{I} - \mathbf{F}^T\mathbf{C}]\mathbf{A}^{-1} = [\mathbf{I} + \mathbf{A}^{-1}\mathbf{C}^T\mathbf{EC}]\mathbf{A}^{-1} = [\mathbf{I} + \mathbf{A}^{-1}\mathbf{C}^T[\mathbf{B} - \mathbf{CA}^{-1}\mathbf{C}^T]^{-1}\mathbf{C}]\mathbf{A}^{-1} = \mathbf{ZA}^{-1}$$

$$\mathbf{D} = \mathbf{ZA}^{-1} \quad \text{with} \quad \mathbf{Z} \equiv [\mathbf{I} + \mathbf{A}^{-1}\mathbf{C}^T[\mathbf{B} - \mathbf{CA}^{-1}\mathbf{C}^T]^{-1}\mathbf{C}]$$

I have tested these formulas numerically; they give the matrix inverse to within machine precision. The solution of the GLS problem is:

$$\begin{bmatrix} \mathbf{m_1} \\ \mathbf{m_2} \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{bmatrix}$$

We now develop an implementation of this formula, presuming that the solution, $\hat{\mathbf{m}}_1 = \mathbf{A}^{-1}\mathbf{a_1}$ for the previous time steps is known. However, we do not want to explicitly construct any matrix inverses (and especially not $\mathbf{A}^{-1}$) because they may not be sparse. Instead, we want to solve linear equations, say via the conjugate gradient method. We start by defining variables $\mathbf{U}$, $\mathbf{Y}$, and $\hat{\mathbf{m}}_2$, all of which can be calculated by solving linear equations:

$$\text{the } MK \times M \text{ matrix } \mathbf{U}: \quad \mathbf{A}^{-1}\mathbf{C}^T \equiv \mathbf{U} \quad \text{or} \quad \mathbf{AU} = \mathbf{C}^T$$

$$\text{the } M \times MK \text{ matrix } \mathbf{Y}: \quad [\mathbf{B} - \mathbf{CU}]^{-1}\mathbf{C} \equiv \mathbf{Y} \quad \text{or} \quad [\mathbf{B} - \mathbf{CU}]\mathbf{Y} = \mathbf{C}$$

$$\text{the } M \times 1 \quad \text{vector} \quad \hat{\mathbf{m}}_2\text{:} \quad \mathbf{B}^{-1}\mathbf{a}_2 \equiv \hat{\mathbf{m}}_2 \quad \text{or} \quad \mathbf{B}\hat{\mathbf{m}}_2 = \mathbf{a}_2$$

These definitions imply:

$$\mathbf{E} = [\mathbf{B} - \mathbf{CU}]^{-1}$$

$$\mathbf{Z} = [\mathbf{I} + \mathbf{U}Y]$$

$$\mathbf{D} = [\mathbf{I} + \mathbf{U}Y]\mathbf{A}^{-1}$$

$$\mathbf{F}^T\mathbf{a}_2 = -\mathbf{D}\mathbf{C}^T\mathbf{B}^{-1}\mathbf{a}_2 = -\mathbf{D}\mathbf{C}^T\hat{\mathbf{m}}_2$$

Then, we write

$$\mathbf{m}_1 = \mathbf{D}\mathbf{a}_1 + \mathbf{F}^T\mathbf{a}_2 = [\mathbf{I} + \mathbf{U}Y][\hat{\mathbf{m}}_1 - \mathbf{U}\hat{\mathbf{m}}_2]$$

$$\mathbf{m}_2 = \mathbf{F}\mathbf{a}_1 + \mathbf{E}\mathbf{a}_2 = \mathbf{E}[\mathbf{a}_2 - \mathbf{C}\hat{\mathbf{m}}_1]$$

$$\text{or} \quad [\mathbf{B} - \mathbf{CU}]\mathbf{m}_2 = \mathbf{a}_2 - \mathbf{C}\hat{\mathbf{m}}_1$$

I have tested these formulas numerically. The recursion for the solution is then:

Step 1:

Set $K = 1$ and solve $\mathbf{A}\hat{\mathbf{m}}_1 = \mathbf{a}_1$ for $\hat{\mathbf{m}}_1$.

Step 2:

Increment K

Assign the submatrices **B** and **C**

Solve for **U**, **Y**, and $\hat{\mathbf{m}}_2$

Solve for $\mathbf{m}_1$ and $\mathbf{m}_2$, the latter is the real-time solution

Step 3:

$\begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}$ becomes $\hat{\mathbf{m}}_1$

$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix}$ becomes $\mathbf{A}$

Go to Step 2

The down side is that the effort needed to compute **U** and **Y** grows with time.

This procedure can be considered a generalization of Kalman filtering.