CENTRE FOR ECONOMETRIC ANALYSIS
CEA@Cass

http://www.cass.city.ac.uk/cea/index.html

Cass Business School
Faculty of Finance
106 Bunhill Row
London EC1Y 8TZ

Practical Issues with State Space Models with Mixed Stationary and Non-Stationary Dynamics

Thomas A. Doan

CEA@Cass Working Paper Series

WP–CEA–05-2009

Practical Issues with State Space Models with Mixed Stationary and Non-Stationary Dynamics

Thomas A. Doan
Estima

Draft Version November 20, 2009

Abstract

State-space models, and the state-space representation of data, are an important tool for econometric modeling and computation. However, when applied to observed (rather than detrended) data, many such models have a mixture of stationary and non-stationary roots. While Koopman (1997) and Durbin and Koopman (2002) provide "exact" calculations for models with non-stationary roots, these have not yet been implemented in most software. Also, neither the Koopman article nor the Durbin and Koopman book address (directly) the handling of models where a unit root is shared among several series—a frequent occurrence in state-space models derived from DSGE's. This paper provides a unified framework for computing the finite and "infinite" components of the initial state variance matrix which is both flexible enough to handle mixed roots and also faster (even for purely stationary models) than standard methods. In addition, it examines some special problems that arise when the number of unit roots is unknown *a priori*.

1 Introduction

State-space models, and the state-space representation of data, are an important tool for econometric modeling and computation. However, when applied to observed (rather than detrended) data, many such models have a mixture of stationary and non-stationary roots. While Koopman (1997) and Durbin and Koopman (2002) provide "exact" calculations for models with non-stationary roots, these have not yet been implemented in most software. Also, neither the Koopman article nor the Durbin and Koopman book address (directly) the handling of models where a unit root is shared among several series—a frequent occurrence in state-space models derived from DSGE's. This paper provides a unified framework for computing the finite and "infinite" components of the initial state variance matrix which is both flexible enough to handle mixed roots and also faster (even for purely stationary models) than standard methods. In addition, it examines some special problems that arise when the number of unit roots is unknown *a priori*.

We will use the following as our general description of a state-space model:

$$\mathbf{X}_t = \mathbf{A}_t\mathbf{X}_{t-1} + \mathbf{Z}_t + \mathbf{F}_t\mathbf{w}_t \tag{1}$$

$$\mathbf{Y}_t = \mu_t + \mathbf{c}_t'\mathbf{X}_t + \mathbf{v}_t \tag{2}$$

Our main focus will be on the properties of (1), and, in particular, that equation with time-invariant component matrices, $\mathbf{A}$, $\mathbf{z}$, $\mathbf{F}$ and $E\mathbf{W}_t\mathbf{W}'_t$.

In the Unobserved Components (UC) models analyzed extensively in Harvey (1989), the $\mathbf{A}$ matrices have all unit roots; the local level model has a single unit root, the local trend has a double (repeated) unit root, and the local seasonal model has the $S-1$ seasonal roots of 1 (other than 1 itself). The state space representation applied to ARMA models, on the

other hand, has all roots inside the unit circle in order to permit the calculation of the full sample likelihood—the non-stationarity of the ARIMA model is typically handled by differencing the data and applying the state space techniques to that.

When a model is constructed by "adding" independent components, the **A** matrix is formed by concatenating the component **A**'s diagonally. For instance, if the observable is the sum of a local trend plus a AR(2) cycle, the **A** matrix is:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \phi_1 & \phi_2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

This is particularly convenient, because the block diagonality allows the two submodels to be initialized separately.

The execution of the Kalman filter requires a pre-sample mean and variance. The ergodic variance is the solution Σ_0 to:

$$\mathbf{A}\Sigma_0\mathbf{A}' + \Sigma_{\mathbf{W}} = \Sigma_0 \tag{4}$$

where $\Sigma_{\mathbf{W}} = \mathbf{F}\left(E\mathbf{W}_t\mathbf{W}_t'\right)\mathbf{F}'$. Where this exists, it's the ideal starting variance since it is the one initializer that is consistent with the structure of the model. However, it's well known that there is no solution to (4) if **A** has roots on the unit circle. A random sampling of papers has indicated that the most common way to handle this is use an approximate "diffuse" prior, that is, a pre-sample mean of 0 with a covariance matrix which is a diagonal matrix with "large" numbers. While not completely unreasonable, there are a number of numerical problems with this approach, particularly if the estimates of state variances are of interest.

Koopman (1997) and Durbin and Koopman (2002) provide a (relatively)

simple exact method for handling diffuse priors in models with (any type of) unit roots. The textbook also describes a way of handling with a mixture of unit and stationary roots, but offers little guidance as to how to handle this in any situation where the states don't partition nicely as in (3). For instance, their description of the non-stationary ARIMA model uses $y_t, \Delta y_t, \Delta y_{t-1}, ...$ as the states rather than $y_t, y_{t-1}, y_{t-2}, ...$ However, this isn't always practical—in some cases, the state space model is derived from a more complicated problem, and the precise positioning of states may not be well known to the user.

This paper proposes a mechanical method which uses the properties of the $\mathbf{A}$ and $\Sigma_{\mathbf{W}}$ matrices only (thus not requiring any ingenuity on the part of the user) to derive a partially diffuse, partially stationary initialization, which is, in addition, faster than the textbook description for the calculation of the ergodic variance. This is based upon a Schur decomposition of $\mathbf{A}$. First, we'll look at how a diffuse prior works in practice, then we'll look at the types of calculations previously used.

2 Diffuse Prior

In practical state space models, the requirement that the eigenvalues are inside the unit circle is almost never met. In the simplest of the non-stationary models, the local level model, the "solution" for 4 is $\mathrm{var}(\mathbf{X}) = \infty$. Suppose that we use a prior of mean 0 and variance κ where κ is "large". The Kalman updating equation for the mean is[1]

$$x_{1|1} = x_{1|0} + \frac{\sigma^2_{1|0}}{\sigma^2_{1|0} + \sigma^2_v}\left(y_1 - x_{1|0}\right)$$

[1]We use $x_{t|s}$ to denote the mean of x_t given information through s and $\sigma^2_{t|s}$ to be the variance of x_t given s.

With $\sigma^2_{1|0} = \kappa + \sigma^2_w$, for κ large relative to σ^2_w and σ^2_v, this will be approximated by

$$x_{1|1} = x_{1|0} + 1.0\left(y_1 - x_{1|0}\right)$$

or $x_{1|1} = y_1$. This shouldn't come as a surprise; with a very high prior variance, the data will dominate completely. The only data information after one data point is that one value. This calculation isn't particularly sensitive to the precise value of κ as long as it's big enough.

The variance calculation is more problematical. The calculation (in practice) is done by:

$$\sigma^2_{1|1} = \sigma^2_{1|0} - \frac{\left(\sigma^2_{1|0}\right)^2}{\sigma^2_{1|0} + \sigma^2_v}$$

While we can rearrange this (algebraically) to give $\sigma_{1|1} \approx \sigma^2_v$, the calculation will be done in software as the subtraction above. Subtracting two very large and almost equal numbers to produce a small number runs the danger of a complete loss of precision. With the standard 15 digit precision on a modern computer, if κ is greater than $10^{15}\sigma^2_v$, the calculation above will give zero.

So in order for this to give us the desired results by using large values of κ, we need to choose a value which is large enough, but not too large. For this model, we can probably figure out a way to do that. With a more complicated model with multiple variances, some of which might be quite small, we're less likely to find such a value.

Note, by the way, that one suggested strategy is to try several values for κ, and check for sensitivity. While better than just picking an arbitrary number, looking for sensitivity on estimates of the mean will likely overlook the more troublesome calculation of the variance. And the calculation that is *most* affected by approximation error is the state variance for Kalman *smoothing*. Those are highly suspect in any calculation done this

way.[2]

This calculation is effectively identical if the model is, in fact, stationary, but a diffuse prior is being used for convenience. If, instead of the random walk, the underlying state equation is $x_t = \rho x_{t-1} + w_t$ with $|\rho| < 1$, then we still will get $x_{1|1} \approx y_1$ and $\sigma_{1|1} \approx \sigma_v^2$, which is *not* the same as what we would get if we used the ergodic variance.[3]

An alternative to approximating this was provided by Koopman (1997), which is *exact diffuse initialization*. This was later refined in Durbin and Koopman (2002). This is done by doing the actual limits as $\kappa \to \infty$. It is implemented by writing covariance matrices in the form $\Sigma_\infty \kappa + \Sigma_*$ where Σ_∞ is the "infinite" part and Σ_* is the "finite" part. These are updated separately as needed.

This works in the following way for the local level model. The prior is mean 0. The prior covariance is $(1)\kappa + (0)$.[4] Moving to $\sigma_{1|0}$ adds the finite σ_w^2, so we're now at $(1)\kappa + (\sigma_w^2)$. The predictive variance for y_1 further adds σ_v^2, putting us at $(1)\kappa + (\sigma_w^2 + \sigma_v^2)$. The Kalman gain is the multiplier on the prediction error used in updating the state. That's

$$\left((1)\kappa + (\sigma_w^2)\right)\left((1)\kappa + (\sigma_w^2 + \sigma_v^2)\right)^{-1} \tag{5}$$

Inverses can be computed by matching terms in a power series expansion in κ^{-1} as shown in section 7. For computing the mean, all we need are the 1 and κ^{-1} terms, which allows us to write 5 as approximately:

$$\left((1)\kappa + (\sigma_w^2)\right)\left((0) + (1)\kappa^{-1}\right) = (1) + (\sigma_w^2)\kappa^{-1}$$

[2]In a model with more than one diffuse state, the filtered covariance matrix after a small number of data points will still have some very large values. Those are only reduced to "finite" numbers as the result of the the smoothing calculation, which uses information obtained by calculating out to the end of the data set and back, accumulating roundoff error along the way.

[3]With the ergodic variance, $x_{1|1}$ will be a weighted average of 0 and y_1 with weights equal to the ergodic variance and σ_v^2 respectively.

[4]We'll keep the different components in parentheses.

where the approximation in the inverse will produce only an additional term on the order of κ^{-2}. We can *now* pass to the limit (that is, ignore all but the finite term) and get the Kalman gain as 1 (exactly), as we expected.

It probably won't come as a surprise that getting the filtered variance correct requires greater accuracy. For that, we'll need to expand the inverse to the κ^{-2} level. We need this because the updating equation is:

$$\left((1)\kappa + (\sigma_w^2)\right) - \left((1)\kappa + (\sigma_w^2)\right)\left((1)\kappa + (\sigma_w^2 + \sigma_v^2)\right)^{-1}\left((1)\kappa + (\sigma_w^2)\right)$$

and the inverse is sandwiched between two factors, each with a κ. As a result, terms in κ^{-2} in the inverse will produce a finite value when this is multiplied out, and only κ^{-3} and above will be negligible. The second order expansion of the required inverse is

$$\left((0) + (1)\kappa^{-1} - (\sigma_w^2 + \sigma_v^2)\kappa^{-2}\right)$$

The calculation of the filtered variance is most conveniently done as

$$\left(1 - \left((1)\kappa + (\sigma_w^2)\right)\left((1)\kappa + (\sigma_w^2 + \sigma_v^2)\right)^{-1}\right)\left((1)\kappa + (\sigma_w^2)\right) \tag{6}$$

With the second order expansion, the product

$$\left((1)\kappa + (\sigma_w^2)\right)\left((1)\kappa + (\sigma_w^2 + \sigma_v^2)\right)^{-1}$$

produces

$$1 - \sigma_v^2\kappa^{-1} - O(\kappa^{-2}) \tag{7}$$

Plugging that into 6 results in $\sigma_v^2 + O(\kappa^{-1})$. Passing to the limit gives us the result we want.

Note that the calculation here that causes the problem for large, but finite values, is subtracting 7 from 1. If we have a loss of precision there,

we won't get σ_v^2 out; we'll get zero.

While we started with an "infinite" variance, after seeing one data point, we now have $x_{1|1} = y_1$, $\sigma_{1|1} = \sigma_v^2$, which are perfectly reasonable finite values. We can Kalman filter in the standard fashion from there.

3 Previous Algorithms for Calculating the Ergodic Variance

While quadratic in $\mathbf{A}$, (4) is linear in the elements of Σ_0. The standard textbook solution for this[5] is to rearrange it into the linear system:

$$[\mathbf{I} - \mathbf{A} \otimes \mathbf{A}]vec(\Sigma_0) = vec(\Sigma_{\mathbf{W}}) \tag{8}$$

As written, this has some redundant elements, since Σ_0 is symmetric. Still, with those eliminated, it requires solving a linear system with $n(n+1)/2$ components. Since solution of a system of equations is $O(N^3)$ in arithmetic operations, this makes this solution procedure $O(n^6)$. This starts to dominate the calculation time for even fairly modest values of n.[6]

There are some special cases where the calculation of 4 can be simplified considerably. For instance, if the $\mathbf{A}$ matrix takes a form such as:

$$\begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \cdots & \varphi_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \tag{9}$$

which is the transition matrix for an $AR(n)$ process, the bottom $(n-1) \times$

[5] See, for instance, Hamilton (1994), page 378 or Durbin and Koopman (2002), page 112

[6] The most time-consuming calculation in one Kalman filter update is computing $\mathbf{A}\Sigma_{t|t}\mathbf{A}'$ which is $O(n^3)$, repeated T times. Thus, if $n^3 \gg T$, $n^6 \gg n^3 T$ and the greatest time will be spent on the initial variance calculation.

$(n-1)$ corner of $\mathbf{A}\Sigma_0\mathbf{A}'$ is just the top $(n-1)\times(n-1)$ corner of Σ_0. Thus, given solutions for $\sigma_{1j}^{(0)}$, the remainder of the matrix can be filled in by solving recursively $\sigma_{i+1,j+1}^{(0)} = \sigma_{i,j}^{(0)} + \sigma_{i+1,j+1}^{(w)}$. So the problem reduces to a linear system with just the n unknowns $\sigma_{1j}^{(0)}$.

An analogous simplification can be done with the block form of 9 that occurs when the model is a vector autoregression. With m variables and p lags, a direct solution of 8 with $n = mp$ total states becomes prohibitive with all but very small models. The recursive solution cuts this down to solving an mp^2 system of equations, which knocks a factor of m^2 out of the number of calculations. With a large value of p, however, this will still be quite expensive.

In Johansen (2002), the author needed a better algorithm for solving 4 for precisely this reason—the corrections needed the ergodic solution for a state space representation for a VAR. His proposal was to transform 4 by taking a matrix $\mathbf{P}$ such that $\mathbf{PAP}^{-1} = \Lambda$ and $\mathbf{P}\Sigma_w\mathbf{P}^{-1} = \mathbf{D}$, where Λ and $\mathbf{D}$ are diagonal. Pre-multiplying (4) by $\mathbf{P}$ and post-multiplying by $\mathbf{P}^{-1}$ and inserting $\mathbf{P}^{-1}\mathbf{P}$ in two places produces the equation:

$$\mathbf{PAP}^{-1}\mathbf{P}\Sigma_0\mathbf{P}^{-1}\mathbf{PA}'\mathbf{P}^{-1} + \mathbf{P}\Sigma_\mathbf{W}\mathbf{P}^{-1} = \mathbf{P}\Sigma_0\mathbf{P}^{-1}$$

Defining $\Sigma_* = \mathbf{P}\Sigma_0\mathbf{P}^{-1}$ and substituting the reductions for A and Σ_w gives us:

$$\Lambda\Sigma_*\Lambda + \mathbf{D} = \Sigma_*$$

Because Λ is diagonal, this reduces to the set of equations:

$$\sigma_{ij}^{(*)}\lambda_i\lambda_j = d_{ij} + \sigma_{ij}^{(*)} \tag{10}$$

Thus, we can directly compute $\sigma_{ij}^{(*)}$, which will also be a diagonal matrix (since $\mathbf{D}$ is diagonal) and then can recover the desired matrix with $\Sigma_0 =$

$\mathbf{P}^{-1}\Sigma_*\mathbf{P}$. Because finding eigenvalues and vectors of an $n \times n$ matrix is an $O(n^4)$ calculation, this reduces the burden by a factor of $O(n^2)$.[7]

One thing to note is that the requirement that $\mathbf{P}$ be a *joint* diagonalizing matrix isn't really necessary. The solution for (12) is very simple whether $\mathbf{P}\Sigma_w\mathbf{P}^{-1} = \mathbf{D}$ is diagonal or not. One complication with this (depending upon the software used) is that, in general, $\mathbf{P}$ and Λ will be complex matrices, and Σ_* will be Hermitian rather than symmetric. Allowing for complex roots, the λ_j in (12) needs to be conjugated.

A more serious problem is that not all state matrices will be diagonalizable. Although it's a case with unit roots (which will be covered later), the standard local trend model transition matrix

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

isn't diagonalizable, and, in fact, is already in its Jordan form. State space models solved out of DSGE's also frequently have repeated eigenvalues.

4 Schur Decomposition Applied to Stationary Models

We're seeking an alternative which produces the same type of reduction of calculation as Johansen's eigen decomposition, but isn't dependent upon the matrix being diagonalizable. The solution to this is found in the Schur decomposition: every square matrix $\mathbf{A}$ has a (not necessarily unique) representation as $\mathbf{A} = \mathbf{QRQ}^{-1}$ where $\mathbf{Q}$ is a unitary matrix[8] and $\mathbf{R}$ is an upper (or "right") triangular matrix. This again will likely produce complex matrices; to avoid that, it's possible to use the "real" Schur decomposition, which also always exists. For that, $\mathbf{Q}$ is a (real) unitary matrix, and $\mathbf{R}$ is

[7] Because the final step in computing eigenvalues is iterative, the computational complexity isn't precisely known. The $O(n^4)$ is based upon a fixed number of iterations per eigenvalue.

[8] The inverse of a unitary matrix is its conjugate transpose, or, for a matrix with only real elements, its simple transpose.

a real matrix that is block-upper triangular, where the diagonal blocks are either 1×1 (for real eigenvalues) or 2×2 for pairs of complex eigenvalues.

By the same type of changes made above (pre-multiplying (1) by $\mathbf{Q}$ and replacing $\mathbf{X}_{t-1}$ with $\mathbf{Q^{-1}Q}\mathbf{X}_{t-1}$), we can reduce the solution to:

$$\mathbf{QAQ^{-1}Q}\Sigma_0\mathbf{Q}^{-1}\mathbf{QA'Q^{-1}} + \mathbf{Q}\Sigma_{\mathbf{W}}\mathbf{Q}^{-1} = \mathbf{Q}\Sigma_0\mathbf{Q}^{-1} \tag{11}$$

or

$$\mathbf{R}\Sigma_*\mathbf{R}' + \mathbf{Q}\Sigma_{\mathbf{W}}\mathbf{Q}^{-1} = \Sigma_* \tag{12}$$

We now show that there is an efficient recursive algorithm for solving this. Write this in block form:

$$\begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix} \begin{bmatrix} \Sigma_{11}^{(*)} & \Sigma_{12}^{(*)} \\ \Sigma'^{(*)}_{12} & \Sigma_{22}^{(*)} \end{bmatrix} \begin{bmatrix} \mathbf{R}'_{11} & 0 \\ \mathbf{R}'_{12} & \mathbf{R}'_{22} \end{bmatrix} + \begin{bmatrix} \Sigma_{11}^{(d)} & \Sigma_{12}^{(d)} \\ \Sigma'^{(d)}_{12} & \Sigma_{22}^{(d)} \end{bmatrix} = \begin{bmatrix} \Sigma_{11}^{(*)} & \Sigma_{12}^{(*)} \\ \Sigma'^{(*)}_{12} & \Sigma_{22}^{(*)} \end{bmatrix} \tag{13}$$

where we're writing $\Sigma_{\mathbf{D}} = \mathbf{Q}\Sigma_{\mathbf{W}}\mathbf{Q}^{-1}$ which is a known matrix. This expands to the three matrix equations:

$$\mathbf{R}_{11}\Sigma_{11}^{(*)}\mathbf{R}'_{11} + \mathbf{R}_{12}\Sigma'^{(*)}_{12}\mathbf{R}'_{11} + \mathbf{R}_{11}\Sigma_{12}^{(*)}\mathbf{R}'_{11} + \mathbf{R}_{12}\Sigma_{22}^{(*)}\mathbf{R}'_{12} + \Sigma_{11}^{(d)} = \Sigma_{11}^{(*)} \tag{14}$$

$$\mathbf{R}_{11}\Sigma_{12}^{(*)}\mathbf{R}'_{22} + \mathbf{R}_{12}\Sigma_{22}\mathbf{R}'_{22} + \Sigma_{12}^{(d)} = \Sigma_{12}^{(*)} \tag{15}$$

$$\mathbf{R}_{22}\Sigma_{22}^{(*)}\mathbf{R}'_{22} + \Sigma_{11}^{(d)} = \Sigma_{22}^{(*)} \tag{16}$$

From (19), we can see the solution for the bottom right corner doesn't depend upon the remainder of the matrix, and from (18), we see that solving for $\Sigma_{12}^{(*)}$ requires only that we solve for $\Sigma_{22}^{(*)}$ first. Thus the procedure is to solve for the lower right diagonal block, then for the remainder of the columns above it. The solution for $\Sigma_{12}^{(*)}$ can also be done recursively using the triangular structure of $\mathbf{R}_{11}$, starting at the bottom of the column and working up.

(17) can then be reduced to:

$$\mathbf{R}_{12}\Sigma'^{(*)}_{12}\mathbf{R}'_{11} + \mathbf{R}_{11}\Sigma^{(*)}_{12}\mathbf{R}'_{11} + \mathbf{R}_{12}\Sigma^{(*)}_{22}\mathbf{R}'_{12} + \Sigma^{(d)}_{11} = \Sigma^{(*)}_{11} - \mathbf{R}_{11}\Sigma^{(*)}_{11}\mathbf{R}'_{11} \quad (17)$$

which has the same form as the original problem. Thus we repeat this process for the submatrix. In the end, we'll have the solution Σ_* for the altered problem, and can transform back to the original representation with $\mathbf{Q}^{-1}\Sigma_*\mathbf{Q}$.

This can also be done with the real Schur decomposition. In that case, $\mathbf{R}_{22}$ will be either 1×1 or 2×2. If it is 2×2, we can just apply the standard (8) to this subproblem, which has only 3 parameters for solution. The solution in (18) does two columns at a time. The largest subproblem comes in the solution of this at an intersection between two 2×2 blocks. But this is just a standard linear system of 4 equations. Thus, this reduces the solution of the large system to a sequence of 1, 2, 3 and 4 variable subproblems. Since the Schur decomposition is quite a bit less ambitious than the eigen decomposition needed in Johansen's procedure, this actually is quicker than that when both can be applied.

The following gives a comparison of the Schur method with the textbook calculation for various numbers of states. These are in number of calculations per second, and show the rapid deterioration of speed with size for the standard calculation.[9]

n	Schur	Standard
10	21739.13	1562.50
20	2380.95	45.87
30	357.15	4.35
50	41.67	.15
100	2.78	

[9]These were done on a 2GHz Dell Laptop running Windows Vista. The standard method was solved using LU decomposition.

5 Extension to Non-Stationary Models

The extension to non-stationary models is straightforward. If, in (19), $\mathbf{R}_{22}$ has unit eigenvalues, then there is no solution. Zero out the "finite" solution for that set of rows and columns.[10] In a parallel "infinite" matrix, which is zeroed out initially, put an identity matrix at the same location as $\mathbf{R}_{22}$. When done, transform both the finite and infinite matrices back to the original representation by pre-multiplying by $\mathbf{Q}^{-1}$ and post-multiplying by $\mathbf{Q}$

Because of the possibility of round-off error, the test for unit eigenvalues has to allow for some flexibility, that is, the test needs to be whether the roots are bigger than something like .9999999. While necessary, this is not innocuous as we'll see in the next session.

6 Near-Unit Roots

There are some practical problems which arise when a model has near unit roots, or roots that might, or might not be unit.

One of these problems is that there's no continuous definition of the likelihood function when you move from a near unit root to a unit root. The standard recursive definition of the likelihood is based upon the predictive density for the observable. With a large, but finite, variance, that likelihood element is computable, although the log likelihood element will be a large negative number. If the predictive variance is infinite, the likelihood is, theoretically, zero and thus the log likelihood isn't defined.

The only way to handle this is to use the conditional rather than unconditional likelihood. That's done automatically in case a unit root is detected. For a state space model, the conditional likelihood is calculated by not including a certain number of early data points. The number of data

[10] There will be one row and one column if $\mathbf{R}_{22}$ is 1×1 and two of each if it's 2×2.

points skipped needs to be at least large enough so that the observables cover the possible number of unit roots.[11]

The other problem is that the calculation is also discontinuous due to the shift in algorithm once a root gets inside the zone where we treat it as a unit root. We've found this to be the more vexing of the two problems. For instance, in Diebold, Rudebusch, and Aruoba (2006), the authors added three macroeconomic variables to a factor model for the term structure, where each of those new variables individually passes a unit root test, so they all either have unit roots or are quite close. The discontinuity itself isn't as significant a problem as the lack of (true) differentiability that it implies. The arc derivatives, which should converge to the computed directional derivatives, don't. As a result, standard procedures for selecting a step size in hill-climbing methods like BFGS or BHHH will fail to work properly.

In order to finesse this, we found that the simplest solution was to use conditional likelihood with a fully diffuse prior for a certain number of iterations in order to get the estimates relatively close to the proper range. At that point, we switched over to the correctly computed variance. While the fully diffuse prior isn't really correct, it's close enough to use as an improvement on the initial guess values for the parameters.

7 Non-Standard Matrix Calculations

Non-standard matrix inversion was introduced into the statistics literature by Koopman (1997).[12]

[11] If there is only one observable, this would mean if there are r potential unit roots, one should condition on r data points. If there is more than one observable, it could be fewer. In most cases, as long as the number of observables times the number of conditioning data points is at least as large as r, the likelihood will be consistently defined.

[12] *Non-standard analysis* was introduced into mathematics in the 1970's using results from "model theory" to embed the real numbers within a framework that includes "infinite" and "infinitesimal" numbers. It was hoped that this would allow simpler descriptions of limit calculations in real analysis, but never really caught on.

The goal is to compute an exact (limit) inverse of an input (symmetric) matrix of the form

$$\mathbf{A} + \kappa\mathbf{B} \text{ as } \kappa \to \infty$$

In Koopman's case, this was needed for Kalman filtering, where $\mathbf{A} + \kappa\mathbf{B}$ is a covariance matrix with some diffuse (infinite variance) states. It can also be used when we need the inverse of a precision matrix where the precision is infinite (variance is zero) in some directions.

You might think that you could just use a guess matrix of the form $\mathbf{C} + \kappa^{-1}\mathbf{D}$, multiply, match terms and be done quickly. Unfortunately, because matrices generally don't commute, it's very easy to get a left inverse or a right inverse, but not a true inverse. We will start with a case that's more manageable. (The derivation here, by the way, is simpler than that in Koopman's paper). Here, the "infinite" matrix is isolated into an identity block:

$$\begin{aligned} &\left(\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A'}_{12} & \mathbf{A}_{22} \end{bmatrix} + \kappa \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\right) \quad \times \\ &\left(\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C'}_{12} & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D'}_{12} & \mathbf{D}_{22} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & \mathbf{I}_{n-r} \end{bmatrix} + \kappa^{-1} Rem \end{aligned} \tag{18}$$

Since the κ term in the product has to zero out, $\mathbf{C}_{11} = 0$ and $\mathbf{C}_{12} = 0$. Using that and collecting the $O(1)$ terms, we get

$$\left(\begin{bmatrix} \mathbf{D}_{11} & \mathbf{A}_{12}\mathbf{C}_{22} + \mathbf{D}_{12} \\ 0 & \mathbf{A}_{22}\mathbf{C}_{22} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & \mathbf{I}_{n-r} \end{bmatrix}$$

So we get $\mathbf{D}_{11} = \mathbf{I}_r, \mathbf{C}_{22} = \mathbf{A}_{22}^{-1}$ and $\mathbf{D}_{12} = -\mathbf{A}_{12}\mathbf{A}_{22}^{-1}$. $\mathbf{D}_{22}$ is arbitrary; it just ends up in the remainder, so for simplicity we can make it zero. If we

check the reverse multiplication

$$\left(\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1} \begin{bmatrix} \mathbf{I}_r & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & 0 \end{bmatrix}\right)\left(\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix} + \kappa \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\right)$$

we can verify that it also will be the identity + a term in κ^{-1}.

In general, we won't have an input matrix with the nice form in (21). However, for a p.s.d. symmetric matrix $\mathbf{B}$, we can always find a non-singular matrix $\mathbf{P}$ such that $\mathbf{PBP}'$ has that form.[13] So, in general, we can compute the inverse with:

$$(\mathbf{A} + \kappa\mathbf{B})^{-1} = \mathbf{P}'\left(\mathbf{PAP}' + \kappa\mathbf{PBP}'\right)^{-1}\mathbf{P}$$

For an input matrix pair $\{\mathbf{A}, \mathbf{B}\}$, this will produce an output pair $\{\mathbf{C}, \mathbf{D}\}$. How can we use this? If we have a typical pair of mean and precision (β_0 and $\mathbf{H}_0$), and another one which has a mean but infinite precision in some directions (β_∞ and $\kappa\mathbf{H}_\infty$), we can combine the two to get a posterior which has the restriction built-in, without the need to rearrange the entire regression to incorporate the restriction. We use our standard formula for the precision weighted average:

$$(\mathbf{H}_0 + \kappa\mathbf{H}_\infty)^{-1}(\mathbf{H}_0\beta_0 + \kappa\mathbf{H}_\infty\beta_\infty)$$

If we apply the exact inverse, we'll get something of the form:

$$\left(\mathbf{V}_0 + \kappa^{-1}\mathbf{V}_\infty\right)(\mathbf{H}_0\beta_0 + \kappa\mathbf{H}_\infty\beta_\infty)$$

Collecting the finite terms (the term with κ^{+1} has to vanish, the ones with

[13] The simplest to compute is based upon a modified version of the Choleski factorization.

κ^{-1}can be discarded after we multiply out) gives us a posterior with mean

$$(\mathbf{V}_0\mathbf{H}_0\beta_0 + \kappa\mathbf{V}_\infty\mathbf{H}_\infty\beta_\infty)$$

and (singular) covariance matrix

$$\mathbf{V}_0$$

Although it won't be needed for anything we're doing (directly), for some purposes it's necessary to get a more accurate inverse, which goes to a κ^{-2} term and leaves only a κ^{-2} remainder when multiplied out. Again working with the simpler case, our test inverse is

$$\left(\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}'_{12} & \mathbf{C}_{22} \end{bmatrix} + \kappa^{-1}\begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & \mathbf{D}_{22} \end{bmatrix} + \kappa^{-2}\begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}'_{12} & \mathbf{E}_{22} \end{bmatrix}\right)$$

Everything from above goes through, except we now can't make $\mathbf{D}_{22}$ arbitrary. When we multiply out, the κ^{-1} terms are

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}'_{12} & \mathbf{D}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}'_{12} & \mathbf{E}_{22} \end{bmatrix} \textit{ or}$$

$$\begin{bmatrix} \mathbf{A}_{11}\mathbf{D}_{11} + \mathbf{A}_{12}\mathbf{D}'_{12} & \mathbf{A}_{11}\mathbf{D}_{12} + \mathbf{A}_{12}\mathbf{D}_{22} \\ \mathbf{A}'_{12}\mathbf{D}_{11} + \mathbf{A}_{22}\mathbf{D}'_{12} & \mathbf{A}'_{12}\mathbf{D}_{12} + \mathbf{A}_{22}\mathbf{D}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ 0 & 0 \end{bmatrix}$$

The bottom left element in the $\mathbf{AD}$ matrix is zero because of the solution for the higher terms. Since $\mathbf{D}_{22}$ was arbitrary from before, we can now solve for it as $\mathbf{D}_{22} = -\mathbf{A}_{22}\mathbf{A}'_{12}\mathbf{D}_{12}$. With that, we also get $\mathbf{E}_{11} = -(\mathbf{A}_{11}\mathbf{D}_{11} + \mathbf{A}_{12}\mathbf{D}'_{12})$ and $\mathbf{E}_{12} - (\mathbf{A}_{11}\mathbf{D}_{12} + \mathbf{A}_{12}\mathbf{D}_{22})$. $\mathbf{E}_{22}$ is now arbitrary.

The extension of this to the more general $\mathbf{B}$ matrix is as before.

References

DIEBOLD, F. X., G. D. RUDEBUSCH, AND S. B. ARUOBA (2006): "The macroeconomy and the yield curve: a dynamic latent factor approach," *Journal of Econometrics*, 131(1), 309–338.

DURBIN, J., AND S. KOOPMAN (2002): *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.

HAMILTON, J. (1994): *Time Series Analysis*. Princeton: Princeton University Press.

HARVEY, A. C. (1989): *Forecasting, structural time series models and the Kalman filter*. Cambridge: Cambridge University Press.

JOHANSEN, S. (2002): "A Small Sample Correction for the Test of Cointegrating Rank in the Vector Autoregressive Model," *Econometrica*, 70(5), 1929–1961.

KOOPMAN, S. (1997): "Exact Initial Kalman Filtering and Smoothing for Nonstationary Time Series Models," *Journal of American Statistical Association*, 92(6), 1630–1638.