Vector Autoregressions (VARs)

Matthew Cocci

1. Intuition

Let's build up to the intuition of VARs by starting of its relatives, the zero-mean AR(1):

$$y_t = \varphi y_{t-1} + \varepsilon_t$$

The AR(1) tries to predict *this* period's value of some variable, y, given the *last* value of that variable. But we don't need to stop with just last period's lags. We can include as many as we want:

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t$$

This gives us the AR(p) model, with p lags. But again, we don't need to stop there. Suppose that we include the lagged values of *other* variables that might help predict y, like say x:

$$y_t = a_1 y_{t-1} + \cdots + a_p y_{t-p} + b_1 x_{t-1} + \cdots + b_p x_{t-p} + \varepsilon_t \tag{1}$$

And presumably, if x helps predict y, why not use y to predict x? So let's write

$$x_t = c_1 x_{t-1} + \cdots + c_p x_{t-p} + d_1 y_{t-1} + \cdots + d_p y_{t-p} + \eta_t \tag{2}$$

As you can see, we can simplify this a lot. In fact, we might toss Equations 1 and 2 into matrices and vectors to simplify notation. This will be particularly useful when we have lots and lots of variables, parameters, and lags. So we might have

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} c_1 & d_1 & \cdots & c_p & d_p \\ b_1 & a_1 & \cdots & b_p & a_p \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \vdots \\ x_{t-p} \\ y_{t-p} \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \tag{3}$$

And viola. We have a bona fide *vector autoregression*, or VAR.

We arrived at it through successive generalizations of *very* simple intuitive ideas. So if you know time series, and if you know linear algebra, you're home. You know this stuff already. And had you done this 40 years ago and pushed the technique and consequences to their natural implications, you would've gotten a Nobel Prize.

2. Definition and Notation

Now, we'll generalize to non-zero mean process, and we'll fix our notation, which was rather sloppy above. Specifically, we'll let y_t denote an $n \times 1$ vector of observables that we want to predict.

Now define the VAR(p) model as follows:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + u_t \qquad u_t \sim \mathrm{N}_n(0, \Sigma) \tag{4}$$

where ϕ_0 is $n \times 1$ and where Σ and ϕ_i $(i = 1, \ldots, p)$ are $n \times n$. So clearly, y_t can be quite a complicated linear function of it's previous lags and it's components.

Next, let's write Equation 4 in more compact, matrix notation. Specifically, concatenate the ϕ_i column vectors horizontally, and stack the lags of y into one big column vector:

$$y_t = \underbrace{\begin{pmatrix} \phi_0 & \phi_1 & \cdots & \phi_p \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} 1 \\ y_{t-1} \\ \vdots \\ y_{t-p} \end{pmatrix}}_{x_t} + u_t \qquad u_t \sim \mathrm{N}_n(0, \Sigma)$$

$$\Leftrightarrow \qquad y_t = \Phi x_t + u_t \tag{5}$$

where Φ is $n \times (np + 1)$ and x_t is $(np + 1) \times 1$.

3. OLS Estimator

Now that we have the model, let's find the OLS for the parameters in Φ. To do so, we'll take one element (or component) of the vector y_t at a time, so one row of y_t and Φ at a time. So let's minimize the sum of squared errors for row/component i:

$$\min_{\Phi^{(i)}} \sum_{i=1}^{T} \left(y_t^{(i)} - \Phi^{(i)} x_t \right)^2 \tag{6}$$

Now this estimator looks just like standard multivariate OLS regression:

$$0 = \frac{d}{d\Phi^{(i)}} \left\{ \sum_{i=1}^{T} \left(y_t^{(i)} - \Phi^{(i)} x_t \right)^2 \right\} = -2 \sum_{i=1}^{T} x_t' \left(y_t^{(i)} - \Phi^{(i)} x_t \right)$$

$$\Rightarrow \quad \hat{\Phi}^{(i)} = \frac{\sum_{i=1}^{T} x_t' y_t^{(i)}}{\sum_{i=1}^{T} x_t' x_t} = (X'X)^{-1} X' Y^{(i)}$$

where

4. Likelihood Function

Now that we have an easy representation of the our VAR(p) model, we move to the likelihood function. By our definition, y_t, conditional on x_t (the lags $y_{t-1}, \ldots, y_{t-p}$), happens to be normally distributed:

$$\begin{aligned} & p(y_t \mid x_t, \Phi, \Sigma) \sim \mathrm{N}_n(\Phi x_t,\ \Sigma) \\ \Rightarrow \quad & p(y_t \mid x_t, \Phi, \Sigma) \propto |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} \left(y_t - \Phi x_t\right)' \Sigma^{-1} (y_t - \Phi x_t) \right\} \end{aligned} \tag{7}$$

Now we use the trick from the trace section in the appendix to rewrite the previous line

$$\Rightarrow \quad p(y_t \mid x_t, \Phi, \Sigma) \propto |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} \mathrm{tr} \left[\Sigma^{-1} (y_t - \Phi x_t) \left(y_t - \Phi x_t\right)' \right] \right\} \tag{8}$$

5. Joint Density Function

Now, we construct the *joint density function* for the entire series, $Y_{1:T}$, where

$$Y_{t_0:t_1} = (y_{t_0}, \dots, y_{t_1})$$

Therefore, conditional on a presample $y_{-p+1}, \dots, y_0$, the jdf is written

$$\begin{aligned}
p(Y_{1:T} \mid Y_{-p+1:0}, \Phi, \Sigma) &= \prod_{t=1}^{T} p(y_t \mid Y_{-p+1:t-1}, \Phi, \Sigma) && (9) \\
&= \prod_{t=1}^{T} p(y_t \mid Y_{t-p:t-1}, \Phi, \Sigma) && (10) \\
&\propto \prod_{t=1}^{T} |\Sigma|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}\left[\Sigma^{-1} (y_t - \Phi x_t)\,(y_t - \Phi x_t)' \right] \right\}
\end{aligned}$$

where we could jump from Equation 9 to 10 since the density of y_t depends only on the previous p lags, not the entire history up until t. And given that the trace of a sum of two matrices is the sum of the traces, we can simplify the jdf

$$\begin{aligned}
p(Y_{1:T} \mid Y_{-p+1:0}, \Phi, \Sigma) &\propto |\Sigma|^{-\frac{T}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}\left[\Sigma^{-1} \sum_{t=1}^{T} (y_t - \Phi x_t)\,(y_t - \Phi x_t)' \right] \right\} \\
&\propto |\Sigma|^{-\frac{T}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}\left[\Sigma^{-1} (Y - X\Phi)'(\right] \right\}
\end{aligned}$$

where we define

$$Y = \begin{pmatrix} y_1' \\ \vdots \\ y_T' \end{pmatrix} \qquad X = \begin{pmatrix} x_1' \\ \vdots \\ x_T' \end{pmatrix}$$

A. Trace

Definition: If A is an $n \times n$ matrix, then

$$\text{tr}[A] = \sum_{i=1}^{n} a_{ii} \tag{11}$$

which is the sum of diagnoal elements.

Trace Fact: If X is $m \times n$ and Y is $n \times m$, then

$$\text{tr}[XY] = \text{tr}[YX] \tag{12}$$

This isn't dificult to prove, just tedious.

Useful Trick: Suppose that a is an $n \times 1$ vector and B is a symmetric positive definite $n \times n$ matrix. Then

$$a'Ba \text{ is a scalar}$$

Then, since the trace of a scalar is just eqal to that scalar, we can rewrite

$$\begin{aligned} a'Ba &= tr\left[a'Ba\right] \\ &= tr\left[a'(Ba)\right] \end{aligned}$$

Now if we use Equation 12, taking $X = a'$ and $Y = Ba$, we can carry on from the simplification above to write

$$\begin{aligned} a'Ba &= tr\left[a'(Ba)\right] = tr\left[(Ba)a'\right] \\ &= tr\left[Baa'\right] \end{aligned}$$

B. Kronecker Product and Vec Operator

B.1. Definitions

Suppose we have two matrices, A which is $m \times n$ and B which is $p \times q$. Then the *Kronecker Product* of A and B is

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

which implies that the new matrix is $(mp) \times (nq)$.

Next, the *vec operator* takes any matrix A that is $m \times n$ and stacks to columns on top of each other (left to right) to form a column vector of length mn. Supposing that a_i are column vectors to simplify notation:

$$\begin{aligned} \text{if } A &= \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \qquad a_i \in \mathbb{R}^{n\times 1} \\ \text{then } \mathbf{vec}A &= \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \end{aligned}$$

B.2. Properties with Proofs, Kronecker Product

Property 1 Let A be $m \times n$, B be $p \times q$, C be $n \times r$, and D be $q \times s$. Then

$$(A \otimes B)(C \otimes D) = AC \otimes BD \tag{13}$$

Proof. We start by writing:

$$(A \otimes B)(C \otimes D) = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix} \begin{pmatrix} c_{11}D & \cdots & c_{1r}D \\ \vdots & \ddots & \vdots \\ c_{n1}D & \cdots & c_{nr}D \end{pmatrix}$$

Since the matrix D has the same number of rows as B has columns, we can carry out the multiplication to get

□