

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 :

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

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

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

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

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

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 a bit sloppy above. So let y_t denote an $n \times 1$ vector of observables we want to predict.

Now define the VAR(p) model as follows:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + u_t \quad u_t \sim N_n(0, \Sigma) \quad (4)$$

where ϕ_0 is $n \times 1$ and where Σ and ϕ_i ($i = 1, \dots, p$) 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:

$$y_t = \underbrace{(\phi_0 \quad \phi_1 \quad \cdots \quad \phi_p)}_{\Phi'} \underbrace{\begin{pmatrix} 1 \\ y_{t-1} \\ \vdots \\ y_{t-p} \end{pmatrix}}_{x_t} + u_t \quad u_t \sim N_n(0, \Sigma)$$

$$\Leftrightarrow y_t = \Phi' x_t + u_t \quad (5)$$

where Φ is $(np + 1) \times n$ 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 column/component i :

$$\min_{\Phi'_{\cdot i}} \sum_{i=1}^T \left(y_t^{(i)} - \Phi'_{\cdot i} x_t \right)^2 \quad (6)$$

Now this estimator looks just like standard multivariate OLS regression, resulting in the "the normal equation":

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

$$\Rightarrow \hat{\Phi}_{\cdot i} = \frac{\sum_{t=1}^T x'_t y_t^{(i)}}{\sum_{t=1}^T x'_t x_t} = (X'X)^{-1} X'Y^{(i)}$$

where X is $T \times (np + 1)$ and $Y^{(i)}$ is $T \times 1$ where

$$X = \begin{pmatrix} x'_1 \\ \vdots \\ x'_T \end{pmatrix} \quad Y^{(i)} = \begin{pmatrix} y_1^{(i)} \\ \vdots \\ y_T^{(i)} \end{pmatrix} \quad (7)$$

So now we have the estimator for each *column* of Φ , which allows us to write

$$\hat{\Phi} = (\hat{\Phi}_{.1} \quad \dots \quad \hat{\Phi}_{.n}) = (X'X)^{-1}X'Y \quad (8)$$

where X is as above and Y is $T \times n$, which is a matrix with y'_t stacked on top of each other for $t = 1, \dots, T$. So it's the $Y^{(i)}$ defined above placed next to each other in a row.

This notation we've developed also gives us a convenient way to define the *sum of squared OLS residuals* matrix:

$$\hat{S} = (Y - X\hat{\Phi})'(Y - X\hat{\Phi}) \quad (9)$$

Eventually, when we start doing the Bayesian computations, we'll use this in the multivariate analog of a result we recall from univariate mathematical statistics, both of which are written below:

$$\begin{aligned} \sum_{i=1}^m (y_i - \mu)^2 &= \sum_{i=1}^m (y_i - \hat{\mu})^2 + m(\hat{\mu} - \mu)^2 \\ (Y - X\Phi)'(Y - X\Phi) &= \hat{S} + (\Phi - \hat{\Phi})'X'X(\Phi - \hat{\Phi}) \end{aligned} \quad (10)$$

where anything with “ $\hat{}$ ” denotes the sample estimate/analog.

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}, \dots, y_{t-p}), happens to be normally distributed:

$$\begin{aligned} p(y_t \mid x_t, \Phi, \Sigma) &\sim N_n(\Phi x_t, \Sigma) \\ \Rightarrow p(y_t \mid x_t, \Phi, \Sigma) &\propto |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (y_t - \Phi'x_t)' \Sigma^{-1} (y_t - \Phi'x_t) \right\} \end{aligned} \quad (11)$$

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

$$\Rightarrow p(y_t \mid x_t, \Phi, \Sigma) \propto |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} \text{tr} [\Sigma^{-1} (y_t - \Phi'x_t) (y_t - \Phi'x_t)'] \right\} \quad (12)$$

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

$$\begin{aligned} p(Y_{1:T} \mid Y_{-p+1:0}, \Phi, \Sigma) &= \prod_{t=1}^T p(y_t \mid Y_{-p+1:t-1}, \Phi, \Sigma) = \prod_{t=1}^T p(y_t \mid Y_{t-p:t-1}, \Phi, \Sigma) \\ &\propto \prod_{t=1}^T |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \text{tr} \left[\Sigma^{-1} (y_t - \Phi' x_t) (y_t - \Phi' x_t)' \right] \right\} \end{aligned} \quad (13)$$

where we could make the jump in Equation 13 because 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

$$\begin{aligned} p(Y_{1:T} \mid Y_{-p+1:0}, \Phi, \Sigma) &\propto |\Sigma|^{-\frac{T}{2}} \exp \left\{ -\frac{1}{2} \text{tr} \left[\Sigma^{-1} \sum_{t=1}^T (y_t - \Phi' x_t) (y_t - \Phi' x_t)' \right] \right\} \\ &\propto |\Sigma|^{-\frac{T}{2}} \exp \left\{ -\frac{1}{2} \text{tr} \left[\Sigma^{-1} (Y - X\Phi)' (Y - X\Phi) \right] \right\} \end{aligned} \quad (14)$$

where X and Y are defined as above in Equation 7 and the subsequent section discussion.

Now, we use a few tricks to rewrite the jdf representation in Equation 14. Namely, we'll want to take care of the nastiness in the trace operator, so let's simplify that straight away (letting $\hat{\beta} = \text{vec}(\hat{\Phi})$, a vector of coefficients):

$$\begin{aligned} \text{By Eqn 10:} \quad \text{tr} \left[\Sigma^{-1} (Y - X\Phi)' (Y - X\Phi) \right] &= \text{tr} \left[\Sigma^{-1} \left\{ \hat{S} + (\Phi - \hat{\Phi})' X' X (\Phi - \hat{\Phi}) \right\} \right] \\ &= \text{tr} \left[\Sigma^{-1} \hat{S} \right] + \text{tr} \left[\Sigma^{-1} (\Phi - \hat{\Phi})' X' X (\Phi - \hat{\Phi}) \right] \\ \text{By Eqn 17} \quad &= \text{tr} \left[\Sigma^{-1} \hat{S} \right] + \text{tr} \left[(\Phi - \hat{\Phi})' X' X (\Phi - \hat{\Phi}) \Sigma^{-1} \right] \\ \text{By Eqn 19} \quad &= \text{tr} \left[\Sigma^{-1} \hat{S} \right] + (\beta - \hat{\beta})' \left[\Sigma^{-1} \otimes (X' X) \right] (\beta - \hat{\beta}) \\ \text{By Kronecker Prop. 3} \quad &= \text{tr} \left[\Sigma^{-1} \hat{S} \right] + (\beta - \hat{\beta})' \left[\Sigma \otimes (X' X)^{-1} \right]^{-1} (\beta - \hat{\beta}) \end{aligned}$$

Now substitute this back into the jdf to get

$$\begin{aligned} p(Y_{1:T} \mid Y_{-p+1:0}, \Phi, \Sigma) &\propto |\Sigma|^{-\frac{T}{2}} \exp \left\{ -\frac{1}{2} \text{tr} \left[\Sigma^{-1} \hat{S} \right] \right\} \\ &\quad \times \exp \left\{ -\frac{1}{2} (\beta - \hat{\beta})' \left[\Sigma \otimes (X' X)^{-1} \right]^{-1} (\beta - \hat{\beta}) \right\} \end{aligned} \quad (15)$$

6. Bayesian Analysis

So we saw above how to arrive at a new representation for the jdf in Equation 15, and in this section we'll see why that was useful.

6.1. Flat Prior, Σ Given

This is perhaps the easiest case, where we want to find the distributions for Φ assuming a flat prior for Φ and that Σ is given. This gives us a posterior of

$$p(\Phi | Y, \Sigma) \propto p(Y | \Phi, \Sigma)p(\Phi | \Sigma) \propto p(Y | \Phi, \Sigma) \\ \propto$$

A. Trace

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

$$\text{tr}[A] = \sum_{i=1}^n a_{ii} \quad (16)$$

which is the sum of diagonal elements.

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

$$\text{tr}[XY] = \text{tr}[YX] \quad (17)$$

This isn't difficult 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 equal to that scalar, we can rewrite

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

Now if we use Equation 17, taking $X = a'$ and $Y = Ba$, we can carry on from the simplification above to write

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

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:

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

B.2. Properties

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

$$(A \otimes B)(C \otimes D) = AC \otimes BD \tag{18}$$

Property 2 $(A \otimes B)' = (A' \otimes B')$.

Property 3 $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$.

Property 4 $\text{tr}[A'BCD'] = \text{vec}(A)'(D \otimes B)\text{vec}(C)$.

How about some proofs?

B.3. Proof of Property 4

Of all the properties, this one strikes me the most of “Now what the hell?” There is zero intuition for this that I can see, so let’s try to derive it.

$$\text{tr}[A'BCD'] = \text{vec}(A)'(D \otimes B)\text{vec}(C) \quad (19)$$

Proof. We know that the matrix that results from $A'BCD'$ must be square to apply the trace operator to it. So let’s call that the beast inside the trace operator E and suppose that it is $q \times q$. Now what does that imply about the size of our other matrices?

Well to allow the matrix multiplication to be carried out (i.e. we can only multiply matrix X and matrix Y if the columns of X equal the rows of Y), and to have E be $q \times q$, this forces

$$A \text{ is } m \times q \quad B \text{ is } m \times n \quad C \text{ is } n \times p \quad D \text{ is } q \times p$$

where m , n , p , and q are all left unspecified.

Okay then, let’s get a move on

$$\text{tr}[A'BCD'] = \text{tr}[E] = \sum_{i=1}^q e_{ii}$$

Now suppose define $X = A'B$ and $Y = CD'$, implying that X is $q \times n$ and Y is $n \times q$. We can rewrite the diagonal components of E in terms of the rows of X , the $x_{i\cdot}$, and the columns of Y , $y_{\cdot i}$.

$$\text{tr}[E] = \sum_{i=1}^q e_{ii} = \sum_{i=1}^q x_{i\cdot} y_{\cdot i}$$

Now let’s not forget where the rows and columns of X and Y come from. The i th row of X is the product of the i th row of A' (or the i th column of A) with each subsequent column of B . Similarly, each column of Y is the dot product of each successive row of C with the i th column of D' (or the i th row of D). Mathematically,

$$\begin{aligned} \text{tr}[E] &= \sum_{i=1}^q e_{ii} = \sum_{i=1}^q x_{i\cdot} y_{\cdot i} \\ &= \sum_{i=1}^q a'_{\cdot i} (b_{\cdot 1} \quad \cdots \quad b_{\cdot n}) \begin{pmatrix} c_{1\cdot} \\ \vdots \\ c_{n\cdot} \end{pmatrix} d'_{i\cdot} \end{aligned} \quad (20)$$

$$= \sum_{i=1}^q a'_{\cdot i} B C d'_{i\cdot} \quad (21)$$

where $a'_{\cdot i}$ is the i th column of A , transposed, and $d'_{i\cdot}$ is the i th row of D , transposed.

Alright, where did that get us? Well, let's look at that last expression. Let's suppress the three components to the write of $a'_{\cdot i}$ —those nasty vectors and matrices with b 's, c 's, and d 's in them. We'll do so by just grouping those three terms into a “back-box” term called “ z_i ,” which we'll postpone computing for now. This let's us write

$$\text{tr}[E] = \sum_{i=1}^q a'_{\cdot i} z_i \quad (22)$$

where z_i is of size $m \times 1$.

But hey, that's a regular old dot product, so let's rewrite it as such with simple matrix notation:

$$\begin{aligned} \text{tr}[E] &= \sum_{i=1}^q a'_{\cdot i} z_i \\ &= (a'_{\cdot 1} \quad \cdots \quad a'_{\cdot q}) \begin{pmatrix} z_1 \\ \vdots \\ z_q \end{pmatrix} = \text{vec}(A)' Z \end{aligned} \quad (23)$$

Now hopefully you were able to make the jump to Equation 23 yourself once you saw what we were doing. And now we at least have *something* from Equation 19. Now let's try to pin down those z_i terms that helped us get here.

Now, let's pin down that Z matrix and show that, in fact,

$$Z = (D \otimes B) \text{vec}(C) \quad (24)$$

We do so by first using Equations 21, 22, and 23 to infer the following:

$$Z = \begin{pmatrix} z_1 \\ \vdots \\ z_q \end{pmatrix} = \begin{pmatrix} BCd'_{\cdot 1} \\ \vdots \\ BCd'_{\cdot q} \end{pmatrix} \quad (25)$$

Next, let's consider the j th row of the last vector, then rewrite C , expanding it into a different but equivalent representation—a row of column vectors all lined up:

$$\begin{aligned} BCd'_{\cdot j} &= B (c_{\cdot 1} \quad \cdots \quad c_{\cdot p}) d'_{\cdot j} \\ \text{expanding } d'_{\cdot j} &= B (c_{\cdot 1} \quad \cdots \quad c_{\cdot p}) \begin{pmatrix} d_{j1} \\ \vdots \\ d_{jp} \end{pmatrix} \end{aligned}$$

Now distribute B and dot product with the $d'_{\cdot j}$:

$$\begin{aligned} BCd'_{\cdot j} &= (Bc_{\cdot 1}d_{j1} + \cdots + Bc_{\cdot p}d_{jp}) \\ \text{Since the } d_{jm} \text{ are scalars} &= (d_{j1}Bc_{\cdot 1} + \cdots + d_{jp}Bc_{\cdot p}) \end{aligned}$$

Amost there, I promise.

Now stack these BCd'_j terms to get Z as in Equation 25:

$$Z = \begin{pmatrix} z_1 \\ \vdots \\ z_q \end{pmatrix} = \begin{pmatrix} BCd'_1 \\ \vdots \\ BCd'_q \end{pmatrix} = \begin{pmatrix} d_{11}Bc_{.1} + \cdots + d_{1p}Bc_{.p} \\ \vdots \\ d_{q1}Bc_{.1} + \cdots + d_{qp}Bc_{.p} \end{pmatrix} \quad (26)$$

Finally, take out the expanded C term from each row and rewrite the sums in each row as a result of matrix multiplication

$$Z = \begin{pmatrix} d_{11}Bc_{.1} + \cdots + d_{1p}Bc_{.p} \\ \vdots \\ d_{q1}Bc_{.1} + \cdots + d_{qp}Bc_{.p} \end{pmatrix} = \begin{pmatrix} d_{11}B & \cdots & d_{1p}B \\ \vdots & \ddots & \vdots \\ d_{q1}B & \cdots & d_{qp}B \end{pmatrix} \begin{pmatrix} c_{.1} \\ \vdots \\ c_{.p} \end{pmatrix} \quad (27)$$

Or in other words, if you look at the last line closely

$$Z = (D \otimes B)\text{vec}(C)$$

QED (drops mic).

□