Graph Inference on Timeseries Data

D.J. Passey

April 20, 2023

1 Understanding the Limitations of Vector Autoregression

In Stock 2001, the authors point out, "The Taylor [rule] is "backward looking" in the sense that the Fed reacts to past information..., and several researchers have argued that Fed behavior is more appropriately described by forward looking behavior." Though not addressing autoregression directly, this makes an interesting point about how autoregression, and most models are "backward looking" while real world processes involve anticipating the future. The authors go on to explain that the Taylor rule was updated to respond to the autoregressive prediction of what inflation would be in the near future. It is interesting to note that while this new model does incoorperate a notion of predicting the future, mathematically, it is still completely backward looking. Later, the authors mention the following issues:

1. "The standard methods of statistical inference (such as computing standard errors for impulse responses) may give misleading results if some of the variables are highly persistent [8]. Another limitation is that, without modification, standard VARs miss nonlinearities, conditional heteroskedasticity, and drifts or breaks in parameters."

2. "While useful as a benchmark, small VARs of two or three variables are often unstable and thus poor predictors of the future (Stock and Watson [1996])."

3. "However, adding variables to the VAR creates complications, because the number of VAR parameters increases as the square of the number of variables: a ninevariable, four-lag VAR has 333 unknown coefficients (including the intercepts). Unfortunately, macroeconomic time series data cannot provide reliable estimates of all these coefficients without further restrictions."

4. "a common assumption made in structural VARs is that variables like output and inflation are sticky and do respond "within the period" to monetary policy shocks. This seems plausible over the period of a single day, but becomes less plausible over a month or quarter."

2 Working through the assumptions behind VAR in *Multivariate Time Series Analysis* by Tsay

Tsay defines a multivariate time series as

$$\boldsymbol{z}_t = (z_{1t}, \cdots, z_{kt})'$$

The problem of interest is defined as predicting $\boldsymbol{z}_{T+1}$ based on the data $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_T\}$. At this point an abstract model, $g(.)$ is employed to solve the problem via

$$\dot{\boldsymbol{z}}_{T+1} = g(\boldsymbol{z}_T, \boldsymbol{z}_{T-1}, \cdots, \boldsymbol{z}_1)$$

This formulation is great for pedagogical use, but I'll explore some of the details that it obscures. Often, the goal is not simply to predict $\boldsymbol{z}_{T+1}$ but rather to gain insight into the dynamics of the data-generating process. Vector autoregression is a powerful tool, not because of it's predictive power, but because of the statistical guarantees attached to its *explanations* for why the behavior occurred. This distinction is important because prediction tasks are somewhat model agnostic. That is to say that the prediction task doesn't care what kind of model made a given prediction, only if it was a better prediction than another model.

On the other hand, for explanatory tools the details of the model are fundamental. Understanding the assumptions, the structure, and common failure modes of an explanitory model are vital if one is to interpret results relative to a particular phenomenon.

Another way of framing this problem is by assuming an underlying data generating process f. It is hard to go much father than this without assumptions about f. Is it a discrete time system? Is it a continuous system? Is it a hybrid? Can it be described as a system at all?

Maybe we can start by assuming that the time series data of interest belongs to the physical world, (since it can be measured) and therefore, it occurred because of some general system.

Our task is to understand and explain a small piece of the process that created the data. Then we can say that the data-generating process f can tell us what the next data point will be, given the history of all previous data points *along with* the complete state of the world.

Then, we model this data generating process by buildig g, the function that tries to *explain* how the data came to be.

For complex systems, an explanitory model is unlikely to be able to explain very much. In the social sciences, there is so much change in the system of interest, that it might be incorrect to assume that the underlying model g is constant. But, by making simplifications, we can identify and explain certain patterns and processes that occur in the world.

At this point, the model g has no structure. It can be anything. Agent based, a lookup table, a differential equation, a branching process, a difference equation.

Some models are optimized to fit to the data, and based on the optimization technique used, we can infer patterns and structure in the data generating process based on the parameters.

Other models have far fewer parameters and do not need optimized statistical fitting. They offer an explanation by their structure rather than through parameters and p-values.

Vector auto regression begins with a choice to restrict g to a class of dynamical system that is not as expressive as other choices would be. Namely,

$$\dot{\boldsymbol{z}}_{T+1} = \boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 \boldsymbol{z}_T + \boldsymbol{\pi}_2 \boldsymbol{z}_{T-1} + \cdots + \boldsymbol{\pi}_T \boldsymbol{z}_1 \tag{1}$$

where each $\boldsymbol{\pi}_i$ is a $k \times k$ matrix.

Tsay assumes that the time series follows a contimuous multivariate distribution.

2.1 Definition: Weakly Stationary

A timeseries is said to be weakly stationary if $E[\boldsymbol{z}_t] = \boldsymbol{mu} \forall t$ and if $\text{Cov}[\boldsymbol{z}_t] = \Sigma_{\boldsymbol{z}}$

2.2 Definition: Linear Time Series

A k dimensional time series $\boldsymbol{z}_t$ is said to be linear if

$$\boldsymbol{z}_t = \boldsymbol{\mu} + \sum_{i=0}^{\infty} \boldsymbol{\psi}_i \boldsymbol{a}_{t-i} \tag{2}$$

where $\boldsymbol{\mu}$ is a k-dimensional constant vector, $\boldsymbol{\psi}_0 = I$, the $k \times k$ identity matrix, $\boldsymbol{\psi}_i$ $(i > 0)$ are $k \times k$ constant matrices and $\{\boldsymbol{a}_t\}$ is a sequence of idependent identically distributed random vectors with mean zero and positive definite covariance matrix $\Sigma_{\boldsymbol{a}}$

2.3 Wold Decomposition

A stationary, purely stochastic processes $\boldsymbol{z}_t$ can be written as a linear combination of serially uncorrelated processes $\boldsymbol{e}_t$.

2.4 Result

If the coefficient matrices of a linear time series satisfy

$$\sum_{i=0}^{\infty} ||\boldsymbol{\psi}_i|| < \infty$$

then it is is stationary with $E[\boldsymbol{z}_t] = \boldsymbol{\mu}$ and

$$\text{Cov}[\boldsymbol{z}_t] = \sum_{i=0}^{\infty} \boldsymbol{\psi}_i \Sigma_a \boldsymbol{\psi}_i' \tag{3}$$

2.5 Invertibility

A linear time series is invertible if it can be written as

$$\boldsymbol{z}_t = \boldsymbol{c} + \boldsymbol{a}_t + \sum_{j=1}^{\infty} \boldsymbol{\pi}_j \boldsymbol{z}_{t-j} \tag{4}$$

3 Working through Lutkepohl 2005 New Introduction to Multiple Time Series Analysis

The Tsay book was a little too underspecified for an applied mathematician so I switched to Lutkepohl.

3.1 Definition: Multivariate Stochastic Process

Let $(\Omega, \mathcal{F}, \Pr)$ be a probability space where Ω is the event space, $\mathcal{F}$ is a sigma algebra over all subsets of Ω, and $\Pr : \mathcal{F} \to \mathbb{R}$ is a probability measure on $\mathcal{F}$.

We define a *random variable* to be a real valued function $y : \Omega \to \mathbb{R}$ such that for each $c \in \mathbb{R}$, $A_c = \{\omega \in \Omega | \, y(\omega) \leq c\} \in \mathcal{F}$.

Since A_c belongs to $\mathcal{F}$ we can determine it's proportional probability. The function $F_y : \mathbb{R} \to [0, 1]$ defined by $F_y(c) = \Pr(A_C)$, is the *distribution function* of y.

A *K-dimensional random vector*, or a *vector of K random variables* is a function $\boldsymbol{y} : \Omega \to \mathbb{R}^K$ such that for each $\boldsymbol{c} \in \mathbb{R}^k$, $A_{\boldsymbol{c}} = \{\omega | \, y_1(\omega) \leq c_1, ..., y_k(\omega) \leq c_k\} \in \mathcal{F}$.

A *discrete stochastic process* is a real valued function, $y : Z \times \Omega \to \mathbb{R}$ where Z is countable and for each $t \in Z$, $y(t, \omega)$ is a random variable.

A *multivariate stochastic process* is a function

$$\boldsymbol{y} : Z \times \Omega \to \mathbb{R}^K$$

such that for each fixed $t \in Z$, $\boldsymbol{y}(t, \omega)$ is a K-dimensional random vector. For simplicity we will denote this as $\boldsymbol{y}_t$.

3.2 First Two Moments of a Univariate Stochastic Process

$$\begin{aligned} E(y_t) &= \mu_t \\ E[(y_t - \mu_t)^2] & \\ E[(y_t - \mu_t)(y_s - \mu_s)] & \end{aligned}$$

3.3 Data Generating Process

By specifying $\omega = \omega_0$ we can study a particular realization of $\boldsymbol{y}_t(t, \omega)$. It can therefore be thought of as a function $\boldsymbol{y}(t, \omega_0) : Z \to \mathbb{R}^K$. The underlying stochastic process is said to have generated the multiple time series. It is sometimes called the *data generating process* (DGP).

3.4 Vector Autoregressive Process

A univariate autoregressive process is a stochastic process

$$y_t = \nu + \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + u_t$$

where ν and each α_i are constant real numbers and y_t ... y_{t-p} and u_t are scalar random variables with u_t denoting the prediction error $u_t := y_t - \hat{y}_t$ and u_t uncorrelated with u_s for all $s \neq t$.

Similarly, a multivariate autoregressive process is a stochastic process described by the following equation:

$$\boldsymbol{y}_t = \boldsymbol{\nu} + A_1 \boldsymbol{y}_{t-1} + \cdots + A_p \boldsymbol{y}_{t-p} + \boldsymbol{u}_t$$

Here, we assume that the $\boldsymbol{u}_t$ are identically distributed, zero mean, length K random vectors.

3.5 VAR(p) Process

The VAR model of order p or VAR(p) is the following object

$$\boldsymbol{y}_t = \boldsymbol{\nu} + A_1 \boldsymbol{y}_{t-1} + \cdots + A_p \boldsymbol{y}_{t-p} + \boldsymbol{u}_t$$

where $t \in \mathbb{Z}$, $\boldsymbol{y}$ is a length K random vector, $\boldsymbol{\nu}$ is a fixed length K vector, the A_i are $K \times K$ constant coefficient matrices and $\boldsymbol{u}_t$ is a white noise or innovation process with $E(\boldsymbol{u}_t) = \boldsymbol{0}$, $E(\boldsymbol{u}_t \boldsymbol{u}_t^T) = \Sigma_{\boldsymbol{u}}$ with $\Sigma_{\boldsymbol{u}}$ assumed non-singular, and $E(\boldsymbol{u}_t \boldsymbol{u}_s^T) = \boldsymbol{0}$ when $s \neq t$.

Any VAR(p) process can be rewritten as a VAR(1) process with the transformation,

$$Y_t = \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p} \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} \nu \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad U_t = \begin{bmatrix} \boldsymbol{u}_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$\boldsymbol{A} = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \end{bmatrix}$$

Thus the following proofs are general:

3.6 Result

If the eigenvalues of $\boldsymbol{A}$ are less than one, a VAR(p) can be written in the following form:

$$Y_t = \boldsymbol{\mu} + \sum_{i=0}^{\infty} \boldsymbol{A}^i \boldsymbol{u}_{t-i}$$

3.7 Proof

As show in 3.5, Any VAR(p) process can be written as a VAR(1) process of the form

$$Y_t = \boldsymbol{\nu} + \boldsymbol{A}Y_{t-1} + U_t$$

Applying this recursively gives

$$Y_t = \boldsymbol{\nu} + \boldsymbol{A}(\boldsymbol{\nu} + \boldsymbol{A}Y_{t-2} + U_{t-1}) + U_t \tag{5}$$

$$Y_t = \boldsymbol{\nu} + \boldsymbol{A}\boldsymbol{\nu} + \boldsymbol{A}^2Y_{t-2} + \boldsymbol{A}U_{t-1} + U_t \tag{6}$$

$$Y_t = \boldsymbol{\nu} + \boldsymbol{A}\boldsymbol{\nu} + \boldsymbol{A}^2(\boldsymbol{\nu} + \boldsymbol{A}Y_{t-3} + U_{t-2}) + \boldsymbol{A}U_{t-1} + U_t \tag{7}$$

$$Y_t = (I + \boldsymbol{A} + \boldsymbol{A}^2 + \cdots + \boldsymbol{A}^m)\boldsymbol{\nu} + \boldsymbol{A}^{m+1}Y_{t-(m+1)} + \sum_{i=0} \boldsymbol{A}^i\, U_{t-i}. \tag{8}$$

Since the eigenvalues of $\boldsymbol{A}$ are all less than one, the series

$$\sum_{i=0}^{\infty} \boldsymbol{A}^i = (I - \boldsymbol{A})^{-1}$$

Similarly, $\boldsymbol{A}^m \to 0$ as $m \to \infty$.

Therefore, if we take the limit of the sequence above as m goes to infinity we obtain

$$Y_t = \boldsymbol{\mu} + \sum_{i=1}^{\infty} \boldsymbol{A}^i U_{t-i}$$

Where $\boldsymbol{\mu} = (I - \boldsymbol{A})^{-1}\boldsymbol{\nu}$.

3.7.1 Discussion

This means that a realization of a vector autoregressive process can be described as repeated draws from the same multivariate normal distribution (if we assume the white noise is gaussian, which I think we must for the statistical ananlysis to work.)

We take these repeated draws and multiply them by appropriate powers of $\boldsymbol{A}$. This causes the prevalance of a particular random vector increase or decrease given the magnitude of the entries of $\boldsymbol{A}^i$.

We could approximate this by truncating the infinite series at M terms where $||\boldsymbol{A}^M||$ is sufficiently small. Then we generate a sequence of random M vectors and apply powers of $\boldsymbol{A}$ appropriately. To get the next Y_t, we shift the sequence by one and generate a new random vector to fill in the open spot.

This line of thinking led to a very fruitful investigation in the notebook located here.

Additionally, for an *excellent* description of the way to interpret VAR weights see this stack exchange.

4 The Approximated Jacobian as a Tool for Understanding Systems from Time Series Data

The System Dynamics software, *Stella* makes use of an algorithim called "Loops that Matter", developed by William Schoenfield. The premise of the algorithm is that even when we model a system explicitly, the pathways and relationships between variables that contribute *most* to a system's behavior are difficult to understand. A great video of how this looks is available here.

The algorithm proposes a "Loop Score" that measures the extent that each pathway between variables contributes to the bevaior of the model. Each link between variables is scored with the following:

$$\text{LinkScore}(x \to z) = \frac{\partial z}{\partial x}\left|\frac{dx/dt}{dz/dt}\right|$$

For the loops that matter algorithm, the model is an explicit differential equation and therefore, all chains and loops of variable dependencies can be indentified and collected into a set $P_1, P_2, ...P_n$ where each P_i is a collection of links K_j that make up the loop: $P_i = \{K_1, K_2, ..., K_{N_i}\}$.

We assign each loop an unormalized score,

$$\text{RawLoopScore}(P_i) = \Pi_{j=1}^{N_i}\text{LinkScore}(K_j)$$

then rescale:

$$\text{LoopScore}(P_i) = \frac{\text{RawLoopScore}(P_i)}{\sum_{k=1}^{n}\text{RawLoopScore}(P_k)}$$

The authors offer justifications for each of these decisions, but there is a lot of work to to to make sure a measure assesses what you want it to assess. That said, I think there is a lot to be gained from an approach like this.

First, it acknoleges that the "network" underlying a system is dynamic. This goes a step further than any of the methods I've considered. All of them try to condense the relationship between time series down to a single fixed network.

It makes sense why this is done, and it gives us insight into systems but it feels incomplete. It makes an assumption that the network underlying the system is static, and I think that this is fine but the techniques offer no measurement to assess how well the assumption of a static network holds up in the data.

On the other hand, this technique, applied to time series would allow us to see how much the interdepencence between signals changes over time.

Second, the Stella team has already modeled useful visualizations. It was developed with the goal of connecting structure to behavior and the Stella team have worked hard to make visualizations that help interpret how the structure relates to dynamics and how this changes with time.

The challenge is that "Loops that Matter" assumes that the model structure is explicitly known. It uses the model's equations to compute partial derivatives at different points in time. For this technique to work on real data, we would need to estimate partial derivatives.

4.1 Problem Formulation

Let $\boldsymbol{x}(t) = \left[x_1(t), x_2(t), \cdots, x_n(t)\right]^T$ be a vector of endogenous variables and let $\boldsymbol{u}(t) = \left[u_1(t), u_2(t), \cdots, u_m(t)\right]^T$ be a vector of exogenous variables. Assume that the evolution of the endogenous can be described as a (stochastic) function of the current state. (This assumption does not allow time delays unless a time delayed variables is included as one of the exogenous or endogenous variables.)

$$\dot{\boldsymbol{x}}(t) = F(\boldsymbol{x}(t), \boldsymbol{u}(t))$$

Assume that we have N exogenous inputs and the corresponding N orbits generated by this system, $\{\boldsymbol{x}_i(t)\}_{i=1}^N$ and $\{\boldsymbol{u}_i(t)\}_{i=1}^N$, but F is unknown. Given this information, what can be said about the Jacobian of F?

If we don't assume the Jacobian to be linear, but we do assume it to be a (stochastic) continuous function, then what can we say confidently?

I think ideally we would want a Jacobian estimation method that approximated the Jacobian as a mapping rather than fixed matrix and additionally *provided a measure of uncertainty* about the values that it produced.

For example, if many orbits pass in the neighborhood of $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$, then we have a high degree of certainty about $J(\boldsymbol{x}_0, \boldsymbol{u}_0)$ and we should have some metric that expresses this.

In order to evaluate this method, we could initialize system dynamics models, implement their true Jacobians and then generate many time series. We could test our approximated Jacobian against the true jacobian and test our measure of uncertainty against adding noise or certain kinds of orbits.

4.2 Overall Assessment

Overall, I think this would be a huge task. There are so many tricky details. Extracting the jacobian from SD models, deciding which variables to include, and deciding how to sample initial conditions would all be hard, and that doesn't even begin the task of jacobian estimation.

However, I do think this is a good way to think about time series and systems, and not many computational people seem to think about it this way, even though it gets at the heart of what we want to know.

I think it would be a contribution in itself if I built a repository for benchmarking jacobian estimation and only provided a bad method for doing it but opened the space up for other researchers to beat me.

5 Big Picture

I want a method that can take in time series data and tell me the probability that any variable influences any other, along with the direction of influence, positive or negative, and how that influence changes with time.

Tell me the amount of change in one variable that can reasonably be explained by another variable. Point me to the culprit.

Now the tricky part here is that what if there is no change in one variable, but it is the main reason for the change in another variable. For example, my foot is holding the gas pedal down and it is the cause of my acceleration, but you could not infer that because there is no change in my foot position.

What algorithms can give me this information?

One possible route is to use SINDY and build some method of assessing uncertainty.

That is a good algorithm. One that knows when it is likely to be wrong.

I want a dynamic theory discovery algorithm that provides a measure of how related the variables are along with how uncertain it is about the relationship. I want to show that when it is wrong, it knows it is wrong. That the false positive rate is controlled by the p-value.

What I need to do is put together a bunch of models and their simulations along with the target, presumably a network.