Applied Time Series Econometrics

Oxana Malakhovskaya, NRU HSE

November 05, 2019

Estimation

- The parameters of VARMA (exactly as ARMA) models are estimated by maximum likelihood method.
- For the important special case where there are no moving average terms, it is shown that the ML estimates are obtained by OLS.
- In R, we estimate VAR(p) models with the `VAR()` function from the `vars` package.
- But there are two decisions to make before using the software:

```
* How many variables to include $(k)$?
* What should be the order of the model $(p)$?
```

How many variables? Be reasonable!

- The number of coefficients to be estimated in VAR per equation is $1 + pn$, so the total number of parameters to estimate is:

$$\text{Number of parameters } = n(1 + pn)$$

  Example: $n = 5$, $p = 3$ so the total number of parameters to estimate is equal to 80.

The more coefficients to be estimated the larger the estimation error entering the forecast.

Conclusion: In practice it is usual to keep n small and include only variables that are correlated to each other and therefore useful in forecasting each other.

How many lags? Use information criteria!

- Information criteria are commonly used to select the number of lags to be included.
- The criteria for multivariate processes are direct generalisations of the corresponding ones for univarite processes.

$$AIC(p) = \log|\hat{V}(p)| + \frac{2}{T-s}n^2$$

$$HQ(p) = \log|\hat{V}(p)| + \frac{2loglog(T-s)}{T-s}n^2$$

$$SC(p) = \log|\hat{V}(p)| + \frac{log(T-s)}{T-s}n^2,$$

where $\hat{V}(p) = \frac{1}{T}\sum_{t=1}^{T}\hat{\upsilon}_t\hat{\upsilon}_t'$, $s = \max(p_{max}, q_{max})$

- AIC criterion asymptoticaly overestimates the order with positive probability.

$$\hat{p}(SC) \leq \hat{p}(HQ) \leq \hat{p}(AIC)$$

VAR estimation: example

Two series in dataset named “usconsumption”

```
library(fpp)
library(vars)
data(usconsumption, package = "fpp")
head(usconsumption)
VARselect(usconsumption, lag.max=7, type="const")
```

VAR estimation: example(2)

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      1      1      5
##
## $criteria
##                 1          2          3          4
## AIC(n) -1.2682099 -1.2596793 -1.3088392 -1.3253881 -1.33
## HQ(n)  -1.2207737 -1.1806189 -1.1981546 -1.1830794 -1.16
## SC(n)  -1.1514107 -1.0650140 -1.0363077 -0.9749905 -0.90
## FPE(n)  0.2813374  0.2837572  0.2701654  0.2657667  0.26
##                 7
## AIC(n) -1.2758340
## HQ(n)  -1.0386527
## SC(n)  -0.6918380
## FPE(n)  0.2795247
```

VAR estimation: example(3)

```r
VARselect(usconsumption, lag.max = 7,
          type = "const")$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      1      1      5
```

VAR estimation: example(4)

```r
var1 <- VAR(usconsumption, p = 1, type = "const")
summary(var1)
```

VAR Estimation Results:

Endogenous variables: consumption, income

Deterministic variables: const

Sample size: 163

Log Likelihood: -354.304

Roots of the characteristic polynomial:

0.3845 0.3072

Call:

VAR(y = usconsumption, p = 1, type = "const")

VAR estimation: example(5)

Estimation results for equation consumption:

consumption = consumption.l1 + income.l1 + const

	Estimate Std.	Error	t value	Prob
consumption.l1	0.30891	0.08142	3.794	0.00021 ***
income.l1	0.08267	0.06007	1.376	0.17070
const	0.46203	0.07730	5.977	1.43e-08 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.6499 on 160 degrees of freedom

Multiple R-Squared: 0.1379, Adjusted R-squared: 0.1272

F-statistic: 12.8 on 2 and 160 DF, p-value: 6.966e-06

VAR estimation: example(6)

Estimation results for equation income:

income = consumption.l1 + income.l1 + const

	Estimate Std.	Error	t value	Prob
consumption.l1	0.5633	0.1101	5.119	8.73e-07 ***
income.l1	-0.2316	0.0812	-2.853	0.00491 **
const	0.4841	0.1045	4.632	7.45e-06 ***

Signif. codes: 0 $***$ 0.001 $**$ 0.01 $*$ 0.05 . 0.1   1

Residual standard error: 0.8785 on 160 degrees of freedom

Multiple R-Squared: 0.143, Adjusted R-squared: 0.1323

F-statistic: 13.35 on 2 and 160 DF, p-value: 4.343e-06

Model diagnostics

- A range of diagnostic tests is available for checking the model assumptions and properties formally.
- We will discuss:

```
* test for autucorrelation
* test for nonnormality
```

- Are the variables in the system really helpful for forecasting other variables ?

Portmanteau test: idea

- Portmanteau test for residual autocorrelation checks the null hypothesis that there is no remaining residual autocorrelation at lags 1 to h against the alternative that at least one of the autocorelations is nonzero.

$$H_0 : E_t(\upsilon_t \upsilon_{t-i}') = 0, \quad i = 1, \dots, h > p$$
$$H_1 : E_t(\upsilon_t \upsilon_{t-i}') \neq 0 \quad \text{for at least one } i = 1, \dots, h$$

Portmanteau test statistics

Under H_0 :

$$Q_h = T \sum_{j=1}^{h} tr(\hat{C}_j' \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1}) \sim \chi^2(n^2(h-p)),$$

where $\hat{C}_i = \frac{1}{T} \sum_{t=i+1}^{T} \hat{\upsilon}_t \hat{\upsilon}_{t-i}'$

$\hat{u}_t$ are residuals from an estimated $VAR(p)$ model.

For smaller sample sizes and/or values of h that are not sufficiently large, a corrected test statistic is computed as:

$$Q_h^* = T^2 \sum_{j=1}^{h} \frac{1}{T-j} tr(\hat{C}_j' \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1}),$$

This statistics is similar to Ljung-Box statistics. In applying the test it is recommended to try different values of h.

Breush-Godfrey test for autocorrelation: idea

The Breush-Godfrey test for hth order residual autocorrelation assumes a model:

$$\upsilon_t = B_1 \upsilon_{t-1} + \cdots + B_h \upsilon_{t-h} + \varepsilon_t$$

and checks:

$$H_0 : B_1 = \cdots = B_h = 0$$
$$H_1 : B_1 \neq 0 \text{ or } B_2 \neq 0 \text{ or } \cdots \text{ or } B_h \neq 0$$

Breush-Godfrey test for autocorrelation: statistics

For this purpose, two auxiliary models are estimated:

$$\hat{\upsilon}_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + c_0 + c_1 t + B_1 \hat{\upsilon}_{t-1} + \cdots + B_h \hat{\upsilon}_{t-h} + e_t$$
$$\hat{\upsilon}_t = A_1^r y_{t-1} + \cdots + A_p^r y_{t-p} + c_0^r + c_1^r t + e_t^r$$

If we denote the estimated residuals from these auxiliary regressions as $\hat{e}_t$ and $\hat{e}_t^r$, respectively, and residual covariance estimators as:

$$\tilde{\Sigma}_{un} = \frac{1}{T}\sum_{t=1}^{T} \hat{e}_t \hat{e}_t' \qquad \tilde{\Sigma}_{re} = \frac{1}{T}\sum_{t=1}^{T} \hat{e}_t^r \hat{e}_t^{r'},$$

then under H_0:

$$LM_h = T[n - tr\left(\tilde{\Sigma}_{un}\tilde{\Sigma}_{re}^{-1}\right)] \sim \chi^2(hn^2)$$

Autocorrelation tests: code

For implementing the autocorrelation tests in multivariate framework we use `serial.test()` function from the `vars` package.

- With option `type = "PT.asymptotic"` for asymptotic portmanteau statistics

```
serial.test(var1, lags.pt = 10, type = "PT.asymptotic")
```

- With option `type = "PT.adjusted"` for adjusted portmanteau statistics

```
serial.test(var1, lags.pt = 10, type = "PT.adjusted")
```

- With option `type = "BG"` for Breusch-Godfrey statistics

```
serial.test(var1, lags.pt = 10, type = "BG")
```

Autocorrelation tests: example

```r
serial.test(var1, lags.pt = 10, type = "PT.asymptotic")
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var1
## Chi-squared = 55.082, df = 36, p-value = 0.02182
```

```r
serial.test(var1, lags.bg = 10, type = "BG")
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object var1
## Chi-squared = 57.186, df = 40, p-value = 0.03821
```

Autocorrelation tests: example

We can repeat the estimation and autocorrelation tests for models with more lags.

	PT.asymptotic	Breusch-Godfrey
$p = 1$	0.02182	0.03821
$p = 2$	0.02518	0.0404
$p = 3$	0.2219	0.2147

According to portmanteau statistics and Breusch-Godfrey statistics at least three lags are necessary to avoid residual autocorrelation.

Test for non-normality: algorithm(1)

Multivariate test for nomality can be constructed by generalizing Jarque-Bera univariate test.

1. Residual covariance matrix is estimated:

$$\tilde{\Omega}_\upsilon = \frac{1}{T}\sum_{t=1}^{T}(\hat{\upsilon}_t - \bar{\upsilon}_t)(\hat{\upsilon}_t - \bar{\upsilon}_t)' \quad \text{where } \bar{\upsilon}_t = \frac{1}{T}\sum \hat{\upsilon}_t$$

2. Standardized residuals are computed: $\hat{\upsilon}_t^s = \tilde{\Omega}_\upsilon^{-1/2}(\hat{\upsilon}_t - \bar{\upsilon}_t)$
3. Auxiliary vectors are computed:

$$b_1 = (b_{11}, \dots b_{1n}), \text{ with } b_{1k} = \frac{1}{T}\sum_{t=1}^{T}(\hat{\upsilon}_{kt}^s)^3$$

$$b_2 = (b_{21}, \dots b_{2n}), \text{ with } b_{2k} = \frac{1}{T}\sum_{t=1}^{T}(\hat{\upsilon}_{kt}^s)^4$$

Test for non-normality: algorithm(2)

4. Possible test statistics are computed:

$$s_3^2 = \frac{T}{6} b_1' b_1 \sim \chi(n)$$

$$s_4^2 = \frac{T}{24} (b_2 - 3_n)'(b_2 - 3_n) \sim \chi(n),$$

   where 3_n is a n-dimensional vector: $(3, \ldots, 3)'$

5. Multivariate JB statistics is computed:

$$JB_n = s_3^2 + s_4^2 \sim \chi(2n)$$

Test for non-normality: code

To apply the multivariate non-normality test to the data, the `normality.test()` function from `vars` package is used.

```
normality.test(var3, multivariate.only = FALSE)
```

JB-Test (multivariate)

data: Residuals of VAR object var3 Chi-squared = 49.425, df = 4, p-value = 4.761e-10

Skewness only (multivariate)

data: Residuals of VAR object var3 Chi-squared = 11.525, df = 2, p-value = 0.003143

Kurtosis only (multivariate)

data: Residuals of VAR object var3 Chi-squared = 37.9, df = 2, p-value = 5.89e-09

VAR Forecasting

- Forecasting vector processes is completely analogous to forecasting univariate processes.
- Forecasts are generated from VARs recursively and for each variable included into system.

Example: An estmated bivariate ($n = 2$) VAR model with two lags $VAR(2)$ is:

$$
\begin{aligned}
y_{1,t} &= \hat{\mu}_1 + \hat{\phi}_{1,1,1} y_{1,t-1} + \hat{\phi}_{1,1,2} y_{2,t-1} + \hat{\phi}_{2,1,1} y_{1,t-2} + \\
&\quad + \hat{\phi}_{2,1,2} y_{2,t-2} \\
y_{2,t} &= \hat{\mu}_2 + \hat{\phi}_{1,2,1} y_{1,t-1} + \hat{\phi}_{1,2,2} y_{2,t-1} + \hat{\phi}_{2,2,1} y_{1,t-2} + \\
&\quad + \hat{\phi}_{2,2,2} y_{2,t-2}
\end{aligned}
$$

Forecasts(1)

$h = 1$

1. Replace t by $T + 1$ and $y_{i,T+1}$ by $y_{i,T+1|T}$, $i = 1, 2$

$$y_{1,T+1|T} = \hat{\mu}_1 + \hat{\phi}_{1,1,1} y_{1,T} + \hat{\phi}_{1,1,2} y_{2,T} + \hat{\phi}_{2,1,1} y_{1,T-1} + \\ + \hat{\phi}_{2,1,2} y_{2,T-1}$$

$$y_{2,T+1|T} = \hat{\mu}_2 + \hat{\phi}_{1,2,1} y_{1,T} + \hat{\phi}_{1,2,2} y_{2,T} + \hat{\phi}_{2,2,1} y_{1,T-1} + \\ + \hat{\phi}_{2,2,2} y_{2,T-1}$$

Forecasts(2)

$h = 2$

1. Replace t by $T+2$

$$\begin{aligned} y_{1,T+2} &= \hat{\mu}_1 + \hat{\phi}_{1,1,1} y_{1,T+1} + \hat{\phi}_{1,1,2} y_{2,T+1} + \hat{\phi}_{2,1,1} y_{1,T} + \\ &+ \hat{\phi}_{2,1,2} y_{2,T} \\ y_{2,T+2} &= \hat{\mu}_2 + \hat{\phi}_{1,2,1} y_{1,T+1} + \hat{\phi}_{1,2,2} y_{2,T} + \hat{\phi}_{2,2,1} y_{1,T} + \\ &+ \hat{\phi}_{2,2,2} y_{2,T} \end{aligned}$$

2. Replace future values by their forecasts and $y_{i,T+2}$ by $y_{i,T+2|T}$

$$\begin{aligned} y_{1,T+2|T} &= \hat{\mu}_1 + \hat{\phi}_{1,1,1} y_{1,T+1|T} + \hat{\phi}_{1,1,2} y_{2,T+1|T} + \hat{\phi}_{2,1,1} y_{1,T} + \\ &+ \hat{\phi}_{2,1,2} y_{2,T} \\ y_{2,T+2|T} &= \hat{\mu}_2 + \hat{\phi}_{1,2,1} y_{1,T+1|T} + \hat{\phi}_{1,2,2} y_{2,T+1|T} + \hat{\phi}_{2,2,1} y_{1,T} + \\ &+ \hat{\phi}_{2,2,2} y_{2,T} \end{aligned}$$

Forecasting: code and values

To make a forecast for a VAR(p) model we use the `forecast()` function from the `forecast` package:

```
var_fcst <- forecast(var3, h = 8)
var_fcst$forecast$consumption$mean
```

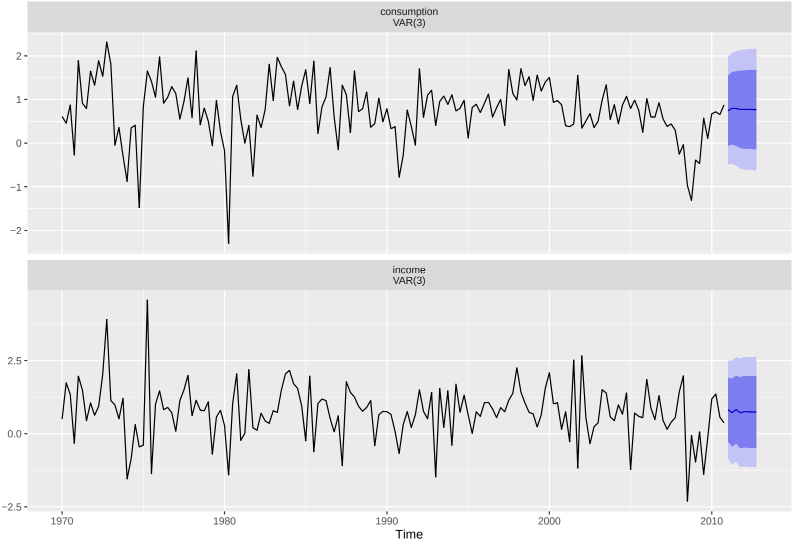
```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2011 0.7421472 0.7980532 0.7921921 0.7742307
## 2012 0.7698384 0.7737721 0.7679183 0.7667572
```

```
var_fcst$forecast$income$mean
```

```
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2011 0.8208025 0.7188650 0.8241248 0.7133820
## 2012 0.7521569 0.7401047 0.7434639 0.7359252
```

Forecasting: code and graph

```r
autoplot(var_fcst) # or just plot(var_fcst)
```



Granger Causality: terminology

- We are going to discuss a conception that is usually called as `Granger causality`. But the title is misleading. It has nothing to do with causes and consequences.
- It is about apossibility to improve a forecast of a variable with lags of another variable.

Granger Causality: idea

In a bivariate VAR(p) model

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \sum_{i=1}^{p} \begin{pmatrix} \phi_{i,1,1} & \phi_{i,1,2} \\ \phi_{i,2,1} & \phi_{i,2,2} \end{pmatrix} \begin{pmatrix} y_{1,t-i} \\ y_{2,t-i} \end{pmatrix} + \upsilon_t$$

y_{2t} is not Granger-causal for y_{1t} if its lags do not appear in the y_{1t} equation.

$$H_0 : \phi_{1,1,2} = \phi_{2,1,2} = \cdots = \phi_{p,1,2} = 0$$
$$H_1 : \text{at least one restriction fails}$$

Analogously, y_{1t} is not Granger-causal for y_{2t} if its lags do not appear in the y_{2t} equation:

$$H_0 : \phi_{1,2,1} = \phi_{2,2,1} = \cdots = \phi_{p,2,1} = 0$$
$$H_1 : \text{at least one restriction fails}$$

Granger Causality: extension

In the trivariate VAR(p) model

$$\begin{pmatrix} y_{1t} \\ y_{2t} \\ y_{3t} \end{pmatrix} = \sum_{i=1}^{p} \begin{pmatrix} \phi_{i,11} & \phi_{i,12} & \phi_{i,13} \\ \phi_{i,21} & \phi_{i,22} & \phi_{i,23} \\ \phi_{i,31} & \phi_{i,32} & \phi_{i,33} \end{pmatrix} \begin{pmatrix} y_{1,t-i} \\ y_{2,t-i} \\ y_{3,t-i} \end{pmatrix} + \upsilon_t$$

information about lags of y_{2t} may still be helpful in forecasting y_{1t} more than one period ahead even if

$$\phi_{1,1,2} = \phi_{2,1,2} = \cdots = \phi_{p,1,2} = 0$$

because lags y_{2t} may be helpful to forecast y_{3t}, and lags y_{3t} in turn may be helpful to forecast y_{1t}.

Instantenous Causality

- A variable y_{2t} is said to be instantanously causal for another time series variable y_{1t} if knowing the value of y_2 in the forecast period helps to improve the forecasts of y_1.
- In a bivariate VAR process, this concept reduces to a property of the model residuals: y_{2t} is instantenously causal for y_{1t} if and only if υ_{1t} and υ_{2t} are correlated.
- The concept is fully symmetric.
- The title is also misleading: the correlation does not imply causation.
- In the case of more than two variables, the problems similar to those encountered for Granger causality emerge: there may be indirect links between variables.

Granger Causality: testing

We apply the Granger causality test with causality() function from the vars package.

```
causality(var3, 'consumption')
causality(var3, 'income')
```

Granger Causality: results(1)

```
## $Granger
##
##  Granger causality H0: consumption do not Granger-cause
##
## data:  VAR object var3
## F-Test = 12.683, df1 = 3, df2 = 308, p-value = 7.731e-0
##
##
## $Instant
##
##  H0: No instantaneous causality between: consumption and
##
## data:  VAR object var3
## Chi-squared = 18.829, df = 1, p-value = 1.43e-05
```

Granger Causality: results(2)

```
## $Granger
##
##  Granger causality H0: income do not Granger-cause consu
##
## data:  VAR object var3
## F-Test = 1.1792, df1 = 3, df2 = 308, p-value = 0.3178
##
##
## $Instant
##
##  H0: No instantaneous causality between: income and cons
##
## data:  VAR object var3
## Chi-squared = 18.829, df = 1, p-value = 1.43e-05
```

Literature

1. Martin, V., Hurn, S., and HarriS,D. (2013) Econometric Modelling with Time Series: Specification, Estimating and Testing, New York, Cambridge University Press
2. Lütkepohl, H., and Krätzig, M. (2004) Applied Time Series Econometrics, New York, Cambridge University Press