Full Information Matrix Maximum Likelihood Estimation

(Hamilton: Chapters 20)

Isabel Casas

Syddansk Universitet

- Johansen's methodology
- Maximum likelihood estimation
- Hipothesis test

VAR model and cointegration

- Granger representation theorem links cointegration to ECM
- Their test only works for one cointegrating vector and the estimation has to be done by OLS
- Søren Johansen links cointegration to a VAR model
- Johansen found a way to both test for the number of cointegrating vectors and then estimate the VAR model using maximum likelihood techniques

Cointegrated VAR (CVAR)

We have the levels VAR(p) model for $\mathbf{y}_t = (y_{1t}, y_{2t}, \ldots, y_{nt})$.

$$\mathbf{y}_t = \boldsymbol{\alpha} + \boldsymbol{\Phi}_1 \mathbf{y}_{t-1} + \ldots + \boldsymbol{\Phi}_p \mathbf{y}_{t-p} + \boldsymbol{\epsilon}_t$$

that can be written as a VAR(p) model on the first differences, also called Vector Error Correction Model (VECM)

$$\boldsymbol{\Delta}\mathbf{y_t} = \boldsymbol{\alpha} + \boldsymbol{\xi}_0 \mathbf{y}_{t-1} + \boldsymbol{\xi}_1 \boldsymbol{\Delta}\mathbf{y}_{t-1} + \ldots + \boldsymbol{\xi}_{p-1} \boldsymbol{\Delta}\mathbf{y}_{t-p+1} + \boldsymbol{\epsilon}_t \quad \boldsymbol{\epsilon}_t \sim IID(\mathbf{0}, \boldsymbol{\Omega})$$

$$\boldsymbol{\xi}_0 = -\left(\mathbf{I}_n - \boldsymbol{\Phi_1} - \boldsymbol{\Phi_2} - \ldots - \boldsymbol{\Phi_p}\right) = -\underbrace{B}_{\mathbf{n \times h}} \underbrace{A'}_{\mathbf{h \times n}}$$

$$\boldsymbol{\xi}_i = -\left(\boldsymbol{\Phi_i} + \boldsymbol{\Phi_{i+1}} + \ldots + \boldsymbol{\Phi_p}\right) \qquad i = 1, \ldots, p-1$$

Each individual variable y_{it} is I(1) and there are h different linear combination of $\mathbf{y}_t$ that are stationary.

Cointegrated VAR (CVAR)

- The estimates are obtained by maximum likelihood.
- If the innovations $\boldsymbol{\epsilon}_t$ are Gaussian, then the loglikelihood of $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_T}$ is:

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_{p-1}, \boldsymbol{\Omega}) = \frac{-Tn}{2}\log(2\pi) - \frac{T}{2}\log|\boldsymbol{\Omega}|$$
$$-\frac{1}{2}\sum_{i=1}^{T}\big[(\boldsymbol{\Delta}\mathbf{y_t} - \boldsymbol{\alpha} - \boldsymbol{\xi}_0\mathbf{y}_{t-1} - \boldsymbol{\xi}_1\boldsymbol{\Delta}\mathbf{y}_{t-1} - \ldots - \boldsymbol{\xi}_{p-1}\boldsymbol{\Delta}\mathbf{y}_{t-p+1})'$$
$$\times\boldsymbol{\Omega}^{-1}(\boldsymbol{\Delta}\mathbf{y_t} - \boldsymbol{\alpha} - \boldsymbol{\xi}_0\mathbf{y}_{t-1} - \boldsymbol{\xi}_1\boldsymbol{\Delta}\mathbf{y}_{t-1} - \ldots - \boldsymbol{\xi}_{p-1}\boldsymbol{\Delta}\mathbf{y}_{t-p+1})\big]$$

- The aim is to find the values that maximise this function

Cointegrated VAR (CVAR)

The Johansen's algorithm consists of three steps to estimate the CVAR model

- Step 1 Calculate a set of auxiliary regressions
- Step 2 Use the residuals of the auxiliary regressions to calculate the sample cannonical correlation matrix. A set of eigenvalues and their correspondent eigenvectors are calculated from these
- Step 3 The eigenvectors are the cointegration relations and they can be used to find the ML estimates

Step 1: Calculate auxiliary regressions

First estimate the VAR(p-1) on the first differences:

$$\boldsymbol{\Delta y_t} = \boldsymbol{\hat{\pi}_0} + \boldsymbol{\hat{\Pi}_1 \Delta y}_{t-1} + \ldots + \boldsymbol{\hat{\Pi}_{p-1} \Delta y}_{t-p+1} + \boldsymbol{\hat{u}_t}$$

where $\boldsymbol{\hat{\Pi}}_i$ contains the OLS coefficients and $\boldsymbol{\hat{u}_t}$ are the OLS residuals

Secondly, estimate the following regressions by OLS:

$$\mathbf{y_{t-1}} = \hat{\boldsymbol{\theta}} + \boldsymbol{\hat{\aleph} \Delta y}_{t-1} + \ldots + \boldsymbol{\hat{\aleph}_{p-1} \Delta y}_{t-p+1} + \boldsymbol{\hat{v}_t}$$

Step 2: Calculate canonical correlations

$$
\begin{aligned}
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{VV}} &= \frac{1}{T}\sum_{i=1}^{T} \hat{\mathbf{v}}_{\mathbf{t}}\hat{\mathbf{v}}_{\mathbf{t}}' \\
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{UU}} &= \frac{1}{T}\sum_{i=1}^{T} \hat{\mathbf{u}}_{\mathbf{t}}\hat{\mathbf{u}}_{\mathbf{t}}' \\
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{UV}} &= \frac{1}{T}\sum_{i=1}^{T} \hat{\mathbf{u}}_{\mathbf{t}}\hat{\mathbf{v}}_{\mathbf{t}}' \\
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{VU}} &= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{UV}}{}'
\end{aligned}
$$

Find the eigenvalues of

$$
\hat{\boldsymbol{\Sigma}}_{\boldsymbol{VV}}{}^{-1}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{VU}}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{UU}}{}^{-1}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{UV}} \tag{1}
$$

and order then $\hat{\lambda}_1 > \hat{\lambda}_2 > \ldots > \hat{\lambda}_h$.

Step 3: Calculate maximum likelihood estimates of parameters

Let $\tilde{\mathbf{a}}_{\mathbf{1}}, \tilde{\mathbf{a}}_{\mathbf{2}}, \ldots, \tilde{\mathbf{a}}_{\mathbf{h}}$ be the eigenvectors associated with the eigenvalues of (1)

$$\mathbf{a} = b_1\tilde{\mathbf{a}}_{\mathbf{1}} + b_2\tilde{\mathbf{a}}_{\mathbf{2}} + \ldots + b_h\tilde{\mathbf{a}}_{\mathbf{h}}$$

for some $(b_1, \ldots, b_h)$.

- This means that each cointegration relation is a linear combination of the eigenvectors
- But there are many of these combinations, so we first normalise the eigenvectors so that $\hat{\mathbf{a}}_{\mathbf{i}} = \tilde{\mathbf{a}}_{\mathbf{i}} / \sqrt{\tilde{\mathbf{a}}_{\mathbf{i}} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{VV}} \tilde{\mathbf{a}}_{\mathbf{i}}}$
- Then we have

$$\hat{\mathbf{A}} = [\hat{\mathbf{a}}_{\mathbf{1}}, \ldots, \hat{\mathbf{a}}_{\mathbf{n}}]$$

Step 3: Calculate maximum likelihood estimates of parameters

The rest of the parameter estimates are:

$$
\begin{aligned}
\hat{\boldsymbol{\xi}}_{\mathbf{0}} =& \hat{\boldsymbol{\Sigma}}_{UV}\hat{\mathbf{A}} \\
\hat{\boldsymbol{\xi}}_{i} =& \hat{\boldsymbol{\Pi}}_{i} - \hat{\boldsymbol{\xi}}_{\mathbf{0}}\hat{\boldsymbol{\aleph}}_{i} \\
\hat{\boldsymbol{\alpha}} =& \hat{\boldsymbol{\pi}}_{\mathbf{0}} - \hat{\boldsymbol{\xi}}_{\mathbf{0}}\hat{\boldsymbol{\theta}} \\
\hat{\boldsymbol{\Omega}} =& \frac{1}{T}\sum_{i=1}^{T}[(\hat{\mathbf{u}}_{\mathbf{t}} - \hat{\boldsymbol{\xi}}_{\mathbf{0}}\hat{\mathbf{v}}_{\mathbf{t}})(\hat{\mathbf{u}}_{\mathbf{t}} - \hat{\boldsymbol{\xi}}_{\mathbf{0}}\hat{\mathbf{v}}_{\mathbf{t}})']
\end{aligned}
$$

SYDDANSK UNIVERSITET

Testing for h cointegrating vectors

$$H_0: h \text{ are the number cointegrating relations}$$
$$H_1: n \text{ are the number of cointegrating relations}$$

- We only need to perform Step 1 and Step 2 to obtain the eigenvectors $\hat{\lambda}_i$
- The number of random walks are $g = n - h$
- The log likelihood ratio test is

$$LRT^1 = -T \sum_{i=h+1}^{n} \log(1 - \hat{\lambda}_i) \sim \text{ Table B.10}$$

- Different cases depending of the value of α and different values depending of g
- This is also called *trace* test

Testing for h cointegrating vectors

$$H_0 : h \text{ are the number cointegrating relations}$$
$$H_1 : h+1 \text{ are the number of cointegrating relations}$$

- We only need to perform Step 1 and Step 2 to obtain the eigenvectors $\hat{\lambda}_i$
- The log likelihood ratio test is

$$LRT^2 = -T\log(1-\hat{\lambda}_{h+1}) \sim \text{ Table B.11 -- distribution}$$

- Different cases depending of the value of $\boldsymbol{\alpha}$ and different values depending of g
- This is also calle *maximum eigenvalue test*

SYDDANSK UNIVERSITET

Example: PPP

The period 1974:Jan – 1989: Oct

```
> load("../data/ppp.rda")
> ppp2= ppp[313:502,]
> ppp.data <- cbind(
+   p=100*log(ppp2[,"PZUNEW"]/ppp2[[1,"PZUNEW"]]),
+   s=-100*log(ppp2[,"EXRITL"]/ppp2[[1,"EXRITL"]]),
+   pstar=100*log(ppp2[,"PC6IT"]/ppp2[[1,"PC6IT"]])
+   )
> y <- as.matrix(ppp.data)
```

Example: PPP

Step 1: Estimate the auxiliary regresions (12 lags) with intercept

```
> lags <- 12
> n <- ncol(y)
> #lag of the first differences
> delta.y.lag <- embed(diff(y),lags)
> X <- delta.y.lag[,-(1:n)]
> T <- nrow(X)
> #Left handside of the equation
> lhs <- cbind( delta.y.lag[,1:n], y[2:(T+1),] )
> #Auxiliary regressions
> aux.lm <- lm( lhs ~ 1 + X, data=list( lhs=lhs, X=X ) )
> #Get the residuals of the OLS regressions all in one command
> uv <- sapply(summary(aux.lm),FUN=function(x) { x$residuals })
> u <- uv[,1:n]
> v <- uv[,(n+1):(2*n)]
```

Example: PPP

Step2: Calculate the canonical correlations

```
> SigmaUU <- 1/T * t(u) %*% u
> SigmaVV <- 1/T * t(v) %*% v
> SigmaUV <- 1/T * t(u) %*% v
> print(SigmaUU)

            Response Y1 Response Y2 Response Y3
Response Y1  0.04334278 -0.02738393  0.01155124
Response Y2 -0.02738393  4.69448944  0.01269838
Response Y3  0.01155124  0.01269838  0.16412603
```

Example: PPP

Step2: Calculate the canonical correlations

```
> print(SigmaVV)

               Response p Response s Response pstar
Response p       385.0404  -345.0220       723.0707
Response s      -345.0220   415.4382      -659.0187
Response pstar   723.0707  -659.0187      1364.0548

> print(SigmaUV)

            Response p Response s Response pstar
Response Y1 -0.3239128  0.3807291    -0.51689656
Response Y2 -0.5290025 -3.8656908    -0.03978276
Response Y3 -1.1131057  0.9821572    -2.24519116
```

Example: PPP

Step2: Find the eigenvalues

```
> eigen.results <- eigen( solve(SigmaVV) %*% t(SigmaUV) %*% solve(SigmaUU) %*% S
> lambda <- eigen.results$values
> LRT1 <- -T*sum(log(1-lambda))
> print(lambda)

[1] 0.10559814 0.03675098 0.02252293

> print(T*log(1-lambda))

[1] -19.864816  -6.664911  -4.054920

> #This the likelihood ratio
> print(LRT1)

[1] 30.58465
```

Example:PPP

We can do the test to know how many cointegrating vectors are there after these two steps.

$$H_0 : h = 0 \text{ vs } H_1 : h = 3$$

Because there is evidence of time trend and we have estimated the regressions with trend then we are in Case 3 of Table B.10. The number of random walks $g = 3$. Therefore the critical value at 5% is 29.5.

The trace test LRT^1 is 38.85>29.5, then the null hypothesis of no cointegration is rejected. Remember this result is different to the result with Engle and Granger tests.

Example:PPP

Now we test

$$H_0 : h = 1 \text{ vs } H_1 : h = 3$$

Then,

$$LRT^1 = (\hat{\lambda}_2 + \hat{\lambda}_3)/T = 10.72$$

The number of random walks is g=2, the critical value for Case 3 is 15.2. Again we reject the null hypothesis concluding there there are 2 relationships.'
Instead if we use the maximum eigenvalue test
$H_0 : h = 1$ vs $H_1 = 2$ then the

$$LRT^2 = -T(\log(1 - \hat{\lambda}_2)) = 6.66 \sim \text{ Table B. 11}$$

The critical value of B.11, case 3 g= 2 at 5% is 14. So then we keep H_0

Example: PPP

Step2: We can use function ca.jo from package urca to do all the above.

```
> options(width=50)
> library(urca)
> ca.jo.results <- ca.jo(y, type = "eigen",  ecdet = "none",
+ K = 12,spec="transitory", season = NULL, dumvar = NULL)
```

Example: PPP

```
Weights W:
(This is the loading matrix)

               p.l1          s.l1      pstar.l1
p.d     -0.03237772 -0.0005168081 -2.917297e-03
s.d     -0.22977578  0.0144007857  6.832073e-05
pstar.d  0.05784111  0.0007968797 -6.205028e-03
```

Example: PPP

Step2: We can use function ca.jo from package urca to do all the above.

```
> summary(ca.jo.results)

######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.10559814 0.03675098 0.02252293

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 2 |  4.05  6.50  8.18 11.65
r <= 1 |  6.66 12.91 14.90 19.19
r = 0  | 19.86 18.90 21.07 25.75

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                 p.l1      s.l1   pstar.l1
p.l1      1.00000000  1.000000  1.0000000
s.l1     -0.03593089 -2.687483 -0.1338774
pstar.l1 -0.55276379 -1.916302 -0.3903290

Weights W:
(This is the loading matrix)
```

Example: PPP

Step 3: We tested and concluded there is only one cointegrating vector which is obtained from the first eigenvalue $\hat{\mathbf{a}}_1$

```
> ahat1 <- eigen.results$vectors[,1]
> #normalisation
> ahat1.tilde <- ahat1 / sqrt( t(ahat1) %*% SigmaVV %*% ahat1 )
> ahat1.normal <- ahat1 / ahat1[1]
> print(ahat1)

[1] -0.87476014  0.03143091  0.48353573

> print(ahat1.tilde)

[1] -0.79006125  0.02838761  0.43671725

> print(ahat1.normal)

[1]  1.00000000 -0.03593089 -0.55276379
```

Example: PPP

```
> vecm.estimation= cajorls(ca.jo.results, r=1)
> #summary(vecm.estimation$rlm)
> vecm.estimation$beta
                ect1
p.l1      1.00000000
s.l1     -0.03593089
pstar.l1 -0.55276379
```