Statistics/Data Analysis

Title

svarih llutkepohl —— Heteroskedasticity-identified SVAR models, Lanne/Lütkepohl (2008) methodology

Syntax

svarih llutkepohl *depvarlist* [*if*] [*in*] **, rgmvar(***rgmvarname***)** [*optional_options*]

options	Description
Model: IH part	
rgmvar(*rgmvarname***)**	identify regimes by contents of variable *rgmvarname*
bconstraints(*constraints_b***)**	apply previously defined *constraints_b* to **B**
beq(*matrix_beq***)**	define and apply to **B** equality constraint matrix *matrix_beq*
bcns(*matrix_bcns***)**	define and apply to **B** cross-parameter constraint matrix *matrix_bcns*
lconstraints(*constraints_l***)**	apply previously defined *constraints_e* to **L**
leq(*matrix_leq***)**	define and apply to **E** equality constraint matrix *matrix_leq*
lcns(*matrix_lcns***)**	define and apply to **E** cross-parameter constraint matrix *matrix_lcns*
Model: VAR part	
lags(*numlist***)**	use lags *numlist* in the underlying VAR
exog(*varlist_exog***)**	use exogenous variables *varlist_exog*
noconstant	suppress constant term
dfk	make small-sample degrees-of-freedom adjustment
small	report small-sample t and F statistics
Reporting and screen output	
level(*#***)**	set confidence level
var	display underlying **var** output
nocnsreport	do not display constraints
notable	do not display estimation results table
display_options	control column formats
Maximization	
evalmode(*modenum***)**	choose between d0, d1, and d2 evaluators; rarely used
glsiter(*glsiternum***)**	perform a maximum number of *glsiternum* GLS iterations
stolerance(*#***)**	convergence criterion for GLS iteration on regime-specific residual covariance matrices
btolerance(*#***)**	convergence criterion for GLS iteration on ML coefficient vector
glstrace	display output of GLS iterations
fixedfrom	always use the same starting values for ML optimizations within the GLS iteration
maximize_options	control the maximization process
coeflegend	display legend instead of statistics

You must **tsset** your data before using **svarih llutkepohl**; see **[TS] tsset**.
varlist_exog may contain time-series operators; see tsvarlist.
depvarlist may NOT contain time-series operators.
See svarih postestimation for features available after estimation.

Description

svarih llutkepohl provides an alternative identification scheme for structural VARs than those implemented in svar. It implements a variant of the the "Identification through Heteroskedasticity" (IH) method as put forth in Lanne/Lütkepohl (2008). For details of the model setup, see the Remarks section. For general information on IH methods and for other IH methods that are available, see svarih.

svarih llutkepohl is limited to two different states of volatility.

Abbreviations, definitions, notation

This help entry uses abbreviations and definitions from svarih.

Options

Model: IH part

rgmvar(*rgmvarname***)** determines the variable whose observations identify the volatility regimes. **svarih llutkepohl** is confined to two states of volatility and they must be numerically encoded in *rgmvarname* by integers 1 and 2. Both values must occur in the estimation sample.

It is not a requirement that a particular regime be contiguous.

bconstraints(*constraints_b***)**, **beq(***matrix_beq***)**, **bcns(***matrix_bcns***)** have the same meaning and can be specified in a similar manner as **bconstraints**, **beq**, **bcns** in **svar**. See the exposition there. They define linear constraints on the contemporaneous shock impact matrix. A difference to **svar** is that **bconstraints()** does accept restrictions across model matrices B and L.

beq(*matrix_beq***)** defines equality constraints. *matrix_beq* usually is an existing matrix but it may also be an expression as in matrix input or a simple matrix function. For example, **beq((.,0\.,.))** and **beq(I(2))** are allowed

bcns(*matrix_bcns***)** defines cross-equation constraints. Argument *matrix_bcns* can be supplied in the same way as *matrix_beq* from option **beq()**.

bconstraints(*constraints_b***)** can define either one. They are defined using the constraint command.

lconstraints(*constraints_l***)**, **leq(***matrix_leq***)**, **lcns(***matrix_lcns***)** work in the same way as **bconstraints**, **beq** and **bcns**. They define linear constraints on the shock modification matrix L which is specific to the IH-LLU model. In IH-LLU, L is a diagonal matrix. **svarih llutkepohl** models the diagonal elements of L as a row vector.

Note that usage of option **lcns()** with positive integers goes against the global identification strategy of IH-LLU, since global identification requires that the elements of L be distinct.

Model: VAR part

lags(*numlist***)** see var / svar.

exog(*varlist_exog***)** see var / svar.

noconstant; see **[R] estimation options**.

dfk see var / svar.

small see var / svar.

Reporting and screen output

level(*#***)**; see **[R] estimation options**.

var specifies that the output from **var** also be displayed. By default, the underlying VAR is fit **quietly**. This option is not available when you replay estimates.

nocnsreport; see **[R] estimation options**.

notable does not display the estimation output table.

display_options: **cformat(***%fmt***)**, **pformat(***%fmt***)**, and **sformat(***%fmt***)**; see **[R] estimation options**.

Maximization

evalmode(*modenum***)** will choose between d0, d1, and d2 evaluators. Rarely used. The default is 2. Can be used to check the numerical robustness of large models.

glsiter(*glsiternum***)** determines the maximum number of GLS iterations. Note that the default is 0, i.e. no GLS iterations at all.

The following options are relevant only if **glsiter(***glsiternum***)**, *glsiternum*>0 is specified:

stolerance(*#***)** and **btolerance(***#***)** are two criteria that must both fulfilled for GLS convergence to be declared.

stolerance(*#***)** calculates a matrix relative difference of the regime-specific reduced-form covariance matrices of the current iteration to the ones from the previous iteration. This criterion is fulfilled if the matrix relative difference is less than #. The default for # is 1e-4.

btolerance(*#***)** calculates a matrix relative difference of the estimated ML coefficient vector of the current iteration to the one from the previous iteration. This criterion is fulfilled if the matrix relative difference is less than #. The default for # is 1e-4.

glstrace outputs the following during each GLS iteration: the estimated coefficient vector, the regime-specific reduced-form covariance matrices, and the ML optimization of the current GLS iteration.

fixedfrom specifies that the same starting values are used for the ML optimization in each GLS iteration. By default, the ML estimates from the previous iteration are used as starting values for the current ML optimization.

maximize_options: **difficult**, **technique(***algorithm spec***)**, **iterate(***#***)**, [**no**]**log**, **trace**, **gradient**, **showstep**, **hessian**, **showtolerance**, **tolerance(***#***)**, **ltolerance(***#***)**, **nrtolerance(***#***)**, **nonrtolerance(***#***)**, and **from(***init_specs***)**; see **[R] maximize**.

coeflegend; see **[R] estimation options**.

Remarks

Remarks are presented under the following headings:

Model equations
GLS iteration to achive ML estimates
Limitations of the current implementation

Model equations

svarih llutkepohl implements the IH-LLU method within a maximum likelihood framework as an extended SVAR B-model. The extension consists of positing a priori knowledge about different regimes of volatility, i.e. time periods during which the structural shocks have different variances. Consequently, a prerequisite for estimation is the specification of a variable that identifies these regimes, which in the following is referred to as the "regime variable".

In terms of a typical structural VAR equation that relates VAR residuals u_t to structural shocks e_t, the model equations of IH-LLU read

(1a) u_t = B*e_t , t is in regime 1
(1b) u_t = B*L^(1/2)*e_t , t is in regime 2

As usual, B models the contemporaneous impact of shocks. L is a diagonal matrix and models the volatility differential between regimes 1 and 2.

If L is constrained to the identity matrix, (1a) and (1b) collapse into a standard SVAR B-model implemented in svar.

GLS iteration to achieve ML estimates

Estimation in **svarih llutkepohl** is based on the optimization of a likelihood function where the VAR-parameters have been concentrated out. The VAR parameters are estimated by equation-by-equation OLS. The resulting parameter estimates are consistent but neither efficient nor do they conincide with ML estimates, due to the assumed existence of different volatility regimes. As a consequence, maximizing the concentrated likelihood will not result in ML estimates of the parameters of interest (the elements of B and L). Lanne/Lütkepohl (2008) therefore propose an iterated feasible GLS procedure which converges to ML estimates. In **svarih llutkepohl**, this GLS iteration is invoked through usage of option **glsiter**. Note that its default value is zero, so if you do not use it, no GLS iterations are performed. In this case, the concentrated likelihood is based on regular VAR lag coefficient estimates. The reason for this implementation is that it is helpful to first check whether ML converges outside of GLS iterations. If this is ensured, re-run the same specification with GLS iterations.

The sequence of calculations is as follows:

1. First, VAR parameter estimates of the lag coefficients are obtained using equation-by-equation OLS. The regime-specific covariance matrices of the implied residuals for regimes 1 and 2 are calculated.
2. The concentrated likelihood is maximized and estimates of B and L are obtained.
   If you omit option **glsiter** or if you explicitly specify **glsiter(0)**, the resulting estimates are returned. If you specified **glsiter(***glsiternum***)**, *glsiternum*>0, the GLS iteration is started.

   GLS iteration
3. Based on the ML estimates of B and L, the regime-specific reduced-form covariance matrices are calculated as B*B' (regime 1) and B*L*B' (regime 2).
4. Based on these two matrices, the implied ML VAR lag coefficients are re-estimated through a non-iterated feasible GLS estimator.
5. Based on the GLS-VAR coefficients, the regime-specific covariance matrices of the implied residuals are calculated.
6. The regime-specific residual covariance matrices are compared to the corresponding matrices of the previous iteration (to the ones of step 1 in the first iteration). Starting in iteration 2, the current ML coefficient vector is also compared to the one from the previous iteration. If all matrices pass a convergence criterion, GLS convergence is declared, and a final ML optimization for B and L is performed based on the GLS-VAR lag coefficient estimates. If convergence was not achieved, the same ML optimization is performed, but subsequently the estimates B and L matrices are again used to start over at step 3.

This procedure has the following implications:

- With **glsiter(0)**, the estimates returned are based on regular VAR lag coefficients. **e()** will contain the lag coefficient estimates in e(b_var) and their covariances in e(V_var).
- With **glsiter(***glsiternum***)**, *glsiternum*>0, the estimates returned are based on the VAR-GLS estimates. In addition to e(b_var) and e(V_var), **e()** will contain the relevant lag coefficient estimates in e(b_vargls) and their covariances in e(V_vargls).
- Estimates are trustworthy if 1) *glsiternum*>0 has been used and 2) ML declared convergence and 3) GLS convergence has been declared.

Limitations of the current implementation

The current implementation is subject to the following limitations:
- IH-LLU can equally be set up as an A-model instead of a B-model. This has not been implemented.
- While one main feature of IH-LLU is global identification of paramters (up to sign reversals of columns of model matrices), a local identification check would still be desirable. A sufficient condition for the global identification is that the elements of L be distinct. In some applications it may nevertheless be necessary to constrain some of the elements of L to be identical and in such situations a check for local identification would be of value. This check is lacking from the current implementation.
- While the check for global indentification also works for more than two regimes, **svarih llutkepohl** can currently only accomodate only two regimes.
- **svarih llu** cannot re-estimate the monetary policy models of Lanne/Lütkepohl (2008), where IH-LLU was proposed. This is because the models of the paper are set up in terms of parameters that underly the elements of B, and because there is no linear mapping between the two sets of parameters.

Examples

The following example illustrates the mechanics of IH-LLU. It does not discuss the important issues of the interpretation of the shocks. It focuses on the mechanics. It builds on the example given in **svar** so you can compare the **svarih llu** setup to the one of **svar**.

Throughout this example section, we store estimated results in Stata's estimation results catalogue for later access. The utility svarih examples allows you to easily re-generate these estimates at any point.

We first load the data set and define constraints. Then we estimate an **svar** model that is similar to the one of the examples section of **[TS] svar**, except that we choose to not restrict the estimation sample here, and that we model contemporaneous relationships strictly through the B-matrix.

```
. webuse lutkepohl2
. matrix aeq = I(3)
. matrix beq = (.,0,0 \ .,.,0 \ .,.,.)
. svar dln_inv dln_inc dln_consump, aeq(aeq) beq(beq)
. est store llu_svar
```

In order to move to IH-LLU, we first must define a regime variable. Furthermore, in order to illustrate the mechanics of IH-LLU, we make the following assumptions: We have prior knowledge that the volatilities of the shocks in our model have changed in the 1970s. We fix a period of differential volatilities of 1974q1. The statement below creates a corresponding regime variable that contains values of 1 and 2, as required by **svarih llu**.

```
. gen byte rgmvar = (qtr>=tq(1974q1)) + 1
```

It is worth reiterating that the occurence of regimes can be modeled to be much more complicated than this. Any sequence and any multiplicity of regimes, with any occurence of gaps in the data, are allowed. If there are enough observations, **svarih llu** will produce estimates.

IH-LLU has a diagonal multiplicative shock modification matrix L, which we decide to leave unconstrained. We continue to work with a lower triangular B-matrix.

```
. svarih llu dln_inv dln_inc dln_consump , rgmvar(rgmvar) beq(beq)
. est store llu_first
```

The global identification check fails: The minimum pairwise Wald statistic that tests for equality among L-elements provides no evidence that the elements of the L-matrix are distinct. In such cases, a local identification check would come in handy, but this is currently not implemented in **svarih llu**.

Luckily, we have used constraints that make the present IH-LLU model equivalent to model bac_first, estimated in svarih bac. It can be shown that the estimates of the two models coincide. Since **svarih bac** concluded with a successful local identification check, the same must hold true for the current IH-LLU model.

Still, in the following we will stop paying attention to identification issues, for the sake of illustrating the mechanics of the command. Some of the following specifications would not pass as valid in applied work.

An advantage of IH-LLU is that it has less stringent requirements concerning constraints than SVAR models. For example, it may be possible to leave B entirely unconstrained. We estimate such a model, and test it against a model with a lower triangular B-matrix. To obtain proper ML estimates, both models are run with the **glsiter()** option.

```
. svarih llu dln_inv dln_inc dln_consump , rgmvar(rgmvar) glsiter(100)
    notable nocnsreport
. est store llu_unconstr_gls
. svarih llu dln_inv dln_inc dln_consump , rgmvar(rgmvar) glsiter(100)
    notable nocnsreport beq(beq)
. est store llu_constr_gls
. lrtest llu_constr_gls llu_unconstr_gls , stat
```

The test rejects the constraints at the 5%-level. Let's inspect the unconstrained estimates a little closer

```
. estimates restore llu_unconstr_gls
. svarih , cmat_l
```

Elements two and three of L are very close, and testing for their equality reproduces the minimum pairwise Wald-statistic in the global identification check:

```
. test [l_1_2]_cons=[l_1_3]_cons
```

On the other hand, we can reject the hypotheses that all three elements are equal:

```
. test [l_1_2]_cons=[l_1_3]_cons=[l_1_1]_cons
```

We now impose in our next estimation that elements 2 and 3 of L be equal.

```
. matrix lcns = (.,1,1)
```

Note than the **lcns()** constraints option works differently than the **leq()** option. The above statement says that two elements of L are constrained to be identical. It does not say that they they are equal to one. This would have to be imposed through **leq()**. We may have imposed the condition that the L-elements are equal by using any other positive integer:

```
. matrix lcns = (.,5,5)
. svarih llu dln_inv dln_inc dln_consump , rgmvar(rgmvar) glsiter(100)
    lcns(lcns)
. est store llu_lcns
```

predict after **svarih llutkepohl** generates prediced values, residuals, shocks, and historical decompositions. **dsimih** generates dynamic simulation statistics. For all features available after estimation, see svarih postestimation.

ADDENDUM: The following demonstrates the equivalence of specifications bac_first and llu_first. We compare the (total) contemporaneous impact of shocks on the endogenous variables for each regime:

```
. svarih examples llu_first , store
. svarih examples bac_first , store ereplace
. est table *_first , p

. matrix Ainv = inv(e(A))
. matrix bac_rgm1 = Ainv * e(B)
. matrix bac_rgm2 = Ainv * (e(B)+e(E))
```

```
. qui est restore llu first
. matrix llu rgm1 = e(B)
. qui svarih , cmat bl nopval
. matrix llu rgm2 = r(BL)

. matrix list bac rgm1
. matrix list llu rgm1

. matrix list bac rgm2
. matrix list llu rgm2
```

Saved results

svarih llutkepohl saves the following in **e()**:

Scalars

e(N)	number of observations
e(N_cns)	number of constraints
e(k_eq)	number of equations in **e(b)**
e(k_dv)	number of dependent variables
e(k_aux)	number of auxiliary parameters
e(ll)	log likelihood
e(N_gaps)	number of gaps in the sample
e(k_var)	number of coefficients in VAR
e(k_eq_var)	number of equations in underlying VAR
e(k_dv_var)	number of dependent variables in underlying VAR
e(df_eq_var)	average number of parameters in an equation
e(df_r_var)	if **small**, VAR residual degrees of freedom
e(mlag)	highest lag in VAR
e(tmin)	first time period in the sample
e(tmax)	maximum time
e(rank)	rank of **e(V)**
e(ic_ml)	number of iterations in the final ML optimization
e(rc_ml)	return code from **ml**
e(converged_ml)	**1** if **ml** declared convergence, **0** otherwise
e(converged_gls)	**1** if the GLS iteration declared convergence, **0** otherwise
e(ic_gls)	number of GLS iterations performed
e(glsiter)	maximium # of GLS iterations allowed in command execution
e(numregimes)	number of regimes marked by *regimevar*
e(Wald_min)	minimum value of the pairwise statistics of Wald tests for equality of elements of L
e(Wald_p)	p-value associated with e(Wald_min)

Macros

e(cmd)	**svarih**
e(method)	**LLutkepohl**
e(version)	version number of command
e(cmdline)	command as typed
e(lags)	lags in model
e(depvar)	names of dependent variables
e(rgmvar)	name of variable that identifies regimes
e(exog)	names of exogenous variables, if specified
e(nocons)	noconstant, if **noconstant** specified
e(cns_b)	comprehensive list of constraints on B
e(cns_l)	comprehensive list of constraints on L
e(dfk_var)	alternate divisor (**dfk**), if specified
e(small)	**small**, if specified
e(tsfmt)	format of timevar
e(timevar)	name of timevar
e(title)	title in estimation output
e(predict)	program used to implement **predict**
e(from)	contents of maximization option **from**, if specified as a string
e(mlopts)	maximization options used
e(regimes)	regime encodings that occur in sample
e(regimes_Ns)	number of observations in each regime
e(glsopts)	GLS-related options supplied to command by user

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(Sigma)	Residual covariance matrix of underlying VAR
e(Sigma_rgm#)	Residual covariance matrix for obs of regime #, based on the underlying GLS-VAR (if **glsiter(**#**)**, #>0) or of the underlying VAR
e(V)	variance-covariance matrix of the estimators
e(b_var)	coefficient vector of underlying VAR model
e(V_var)	VCE of underlying VAR model
	if GLS iteration was performed
e(b_vargls)	coefficient vector of underlying GLS-VAR model
e(V_vargls)	VCE of underlying GLS-VAR model
e(beq)	**beq(***matrix***)**, if specified
e(bcns)	**bcns(***matrix***)**, if specified
e(leq)	**leq(***matrix***)**, if specified
e(lcns)	**lcns(***matrix***)**, if specified
e(B)	estimated B matrix
e(L)	estimated diagonal of L matrix, returned as a row vector
e(from)	matrix of maximization option **from**, if specified as a matrix
e(Wald)	results of all pairwise Wald tests of equality of elements of L

Functions

e(sample)	marks estimation sample

Author

Daniel C. Schneider, Goethe University Frankfurt, dan_schneider@outlook.com

Acknowledgements

The code of official Stata's **svar** has served as a point of reference throughout the development of **svarih llutkepohl**. Any remaining errors in **svarih llutkepohl** are mine.

References

Lanne, M. and H. Lütkepohl (2008): Identifying Monetary Policy Shocks via Changes in Volatility. *Journal of Money, Credit and Banking*, 40 (6), 1131-1149.

Also see

Help: **[TS] svar**, **svarih**, **svarih bac**, **svarih bfa**, **svarih postestimation**, **svarih cmat**, **dsimih**