Title

stata.com

sspace — State-space models

Description

`sspace` estimates the parameters of linear state-space models by maximum likelihood. Linear state-space models are very flexible and many linear time-series models can be written as linear state-space models.

`sspace` uses two forms of the Kalman filter to recursively obtain conditional means and variances of both the unobserved states and the measured dependent variables that are used to compute the likelihood.

The covariance-form syntax and the error-form syntax of `sspace` reflect the two different forms in which researchers specify state-space models. Choose the syntax that is easier for you; the two forms are isomorphic.

Quick start

AR(1) model for `y` with unobserved state `u` modeled as lag of itself in the state equation, and requiring the coefficient of `u` constrained to 1 in the observation equation

```
constraint 1 [y]u = 1
sspace (u L.u, state noconstant) (y u, noerror), constraints(1)
```

Dynamic-factor model of the first difference of `y1`, `y2`, and `y3` as linear functions of an unobserved factor that follows a first-order autoregressive process

```
constraint 1 [y1]u  = 1
sspace (u L.u, state noconstant) (d.y1 u) (d.y2 u) (d.y3 u), nolog
```

Menu

Statistics > Multivariate time series > State-space models

Syntax

Covariance-form syntax

```
sspace state_ceq [state_ceq ... state_ceq] obs_ceq [obs_ceq ... obs_ceq]
    [if] [in] [, options]
```

where each *state_ceq* is of the form

```
(statevar [lagged_statevars] [indepvars], state [noerror noconstant])
```

and each *obs_ceq* is of the form

```
(depvar [statevars] [indepvars] [, noerror noconstant])
```

Error-form syntax

```
sspace state_efeq [state_efeq ... state_efeq] obs_efeq [obs_efeq ... obs_efeq]
    [if] [in] [, options]
```

where each *state_efeq* is of the form

```
(statevar [lagged_statevars] [indepvars] [state_errors], state [noconstant])
```

and each *obs_efeq* is of the form

```
(depvar [statevars] [indepvars] [obs_errors] [, noconstant])
```

statevar is the name of an unobserved state, not a variable. If there happens to be a variable of the same name, the variable is ignored and plays no role in the estimation.

lagged_statevars is a list of lagged *statevar*s. Only first lags are allowed.

state_errors is a list of state-equation errors that enter a state equation. Each state error has the form `e.`*statevar*, where *statevar* is the name of a state in the model.

obs_errors is a list of observation-equation errors that enter an equation for an observed variable. Each error has the form `e.`*depvar*, where *depvar* is an observed dependent variable in the model.

equation-level options	Description
Model	
`state`	specifies that the equation is a state equation
`noerror`	specifies that there is no error term in the equation
`noconstant`	suppresses the constant term from the equation

options	Description
Model	
`covstate(`*covform*`)`	specifies the covariance structure for the errors in the state variables
`covobserved(`*covform*`)`	specifies the covariance structure for the errors in the observed dependent variables
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim` or `robust`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
Advanced	
`method(`*method*`)`	specify the method for calculating the log likelihood; seldom used
`coeflegend`	display legend instead of statistics

covform	Description
`identity`	identity matrix; the default for error-form syntax
`dscalar`	diagonal scalar matrix
`diagonal`	diagonal matrix; the default for covariance-form syntax
`unstructured`	symmetric, positive-definite matrix; not allowed with error-form syntax

method	Description
`hybrid`	use the stationary Kalman filter and the De Jong diffuse Kalman filter; the default
`dejong`	use the stationary De Jong Kalman filter and the De Jong diffuse Kalman filter
`kdiffuse`	use the stationary Kalman filter and the nonstationary large-κ diffuse Kalman filter; seldom used

You must `tsset` your data before using `sspace`; see [TS] **tsset**.
indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.
indepvars and *depvar* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
`by`, `rolling`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**.
`coeflegend` does not appear in the dialog box.
See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Equation-level options

Model

`state` specifies that the equation is a state equation.

`noerror` specifies that there is no error term in the equation. `noerror` may not be specified in the error-form syntax.

`noconstant` suppresses the constant term from the equation.

Options

Model

`covstate(`*covform*`)` specifies the covariance structure for the state errors.

`covstate(identity)` specifies a covariance matrix equal to an identity matrix, and it is the default for the error-form syntax.

`covstate(dscalar)` specifies a covariance matrix equal to σ^2_{state} times an identity matrix.

`covstate(diagonal)` specifies a diagonal covariance matrix, and it is the default for the covariance-form syntax.

`covstate(unstructured)` specifies a symmetric, positive-definite covariance matrix with parameters for all variances and covariances. `covstate(unstructured)` may not be specified with the error-form syntax.

`covobserved(`*covform*`)` specifies the covariance structure for the observation errors.

`covobserved(identity)` specifies a covariance matrix equal to an identity matrix, and it is the default for the error-form syntax.

`covobserved(dscalar)` specifies a covariance matrix equal to $\sigma^2_{\text{observed}}$ times an identity matrix.

`covobserved(diagonal)` specifies a diagonal covariance matrix, and it is the default for the covariance-form syntax.

`covobserved(unstructured)` specifies a symmetric, positive-definite covariance matrix with parameters for all variances and covariances. `covobserved(unstructured)` may not be specified with the error-form syntax.

`constraints(`*constraints*`)`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the estimator for the variance–covariance matrix of the estimator.

`vce(oim)`, the default, causes `sspace` to use the observed information matrix estimator.

`vce(robust)` causes `sspace` to use the Huber/White/sandwich estimator.

Reporting

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

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, and `sformat(%`*fmt*`)`; see [R] **estimation options**.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), and from(*matname*); see [R] **maximize** for all options except from(), and see below for information on from(). These options are seldom used.

from(*matname*) specifies initial values for the maximization process. from(b0) causes sspace to begin the maximization algorithm with the values in b0. b0 must be a row vector; the number of columns must equal the number of parameters in the model; and the values in b0 must be in the same order as the parameters in e(b).

Advanced

method(*method*) specifies how to compute the log likelihood. This option is seldom used.

method(hybrid), the default, uses the Kalman filter with model-based initial values for the states when the model is stationary and uses the De Jong (1988, 1991) diffuse Kalman filter when the model is nonstationary.

method(dejong) uses the Kalman filter with the De Jong (1988) method for estimating the initial values for the states when the model is stationary and uses the De Jong (1988, 1991) diffuse Kalman filter when the model is nonstationary.

method(kdiffuse) is a seldom used method that uses the Kalman filter with model-based initial values for the states when the model is stationary and uses the large-κ diffuse Kalman filter when the model is nonstationary.

The following option is available with sspace but is not shown in the dialog box:

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

Remarks and examples

stata.com

Remarks are presented under the following headings:

An introduction to state-space models

Many linear time-series models can be written as linear state-space models, including vector autoregressive moving-average (VARMA) models, dynamic-factor (DF) models, and structural time-series (STS) models. The solutions to some stochastic dynamic-programming problems can also be written in the form of linear state-space models. We can estimate the parameters of a linear state-space model by maximum likelihood (ML). The Kalman filter or a diffuse Kalman filter is used to write the likelihood function in prediction-error form, assuming normally distributed errors. The quasi–maximum likelihood (QML) estimator, which drops the normality assumption, is consistent and asymptotically normal when the model is stationary. Chang, Miller, and Park (2009) establish consistency and asymptotic normality of the QML estimator for a class of nonstationary state-space models. The QML estimator differs from the ML estimator only in the VCE; specify the vce(robust) option to obtain the QML estimator.

Hamilton (1994a, 1994b), Harvey (1989), and Brockwell and Davis (1991) provide good introductions to state-space models. Anderson and Moore's (1979) text is a classic reference; they produced many results used subsequently. Caines (1988) and Hannan and Deistler (1988) provide excellent, more advanced, treatments.

`sspace` estimates linear state-space models with time-invariant coefficient matrices, which cover the models listed above and many others. `sspace` can estimate parameters from state-space models of the form

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{B}\mathbf{x}_t + \mathbf{C}\boldsymbol{\epsilon}_t$$
$$\mathbf{y}_t = \mathbf{D}\mathbf{z}_t + \mathbf{F}\mathbf{w}_t + \mathbf{G}\boldsymbol{\nu}_t$$

where

$\mathbf{z}_t$ is an $m \times 1$ vector of unobserved state variables;

$\mathbf{x}_t$ is a $k_x \times 1$ vector of exogenous variables;

$\boldsymbol{\epsilon}_t$ is a $q \times 1$ vector of state-error terms, $(q \leq m)$;

$\mathbf{y}_t$ is an $n \times 1$ vector of observed endogenous variables;

$\mathbf{w}_t$ is a $k_w \times 1$ vector of exogenous variables;

$\boldsymbol{\nu}_t$ is an $r \times 1$ vector of observation-error terms, $(r \leq n)$; and

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{F}$, and $\mathbf{G}$ are parameter matrices.

The equations for $\mathbf{z}_t$ are known as the state equations, and the equations for $\mathbf{y}_t$ are known as the observation equations.

The error terms are assumed to be zero mean, normally distributed, serially uncorrelated, and uncorrelated with each other;

$$\boldsymbol{\epsilon}_t \sim N(0, \mathbf{Q})$$
$$\boldsymbol{\nu}_t \sim N(0, \mathbf{R})$$
$$E[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_s'] = \mathbf{0} \text{ for all } s \neq t$$
$$E[\boldsymbol{\epsilon}_t \boldsymbol{\nu}_s'] = \mathbf{0} \text{ for all } s \text{ and } t$$

The state-space form is used to derive the log likelihood of the observed endogenous variables conditional on their own past and any exogenous variables. When the model is stationary, a method for recursively predicting the current values of the states and the endogenous variables, known as the Kalman filter, is used to obtain the prediction error form of the log-likelihood function. When the model is nonstationary, a diffuse Kalman filter is used. How the Kalman filter and the diffuse Kalman filter initialize their recursive computations depends on the `method()` option; see *Methods and formulas*.

The linear state-space models with time-invariant coefficient matrices defined above can be specified in the covariance-form syntax and the error-form syntax. The covariance-form syntax requires that $\mathbf{C}$ and $\mathbf{G}$ be selection matrices, but places no restrictions on $\mathbf{Q}$ or $\mathbf{R}$. In contrast, the error-form syntax places no restrictions $\mathbf{C}$ or $\mathbf{G}$, but requires that $\mathbf{Q}$ and $\mathbf{R}$ be either diagonal, diagonal-scalar, or identity matrices. Some models are more easily specified in the covariance-form syntax, while others are more easily specified in the error-form syntax. Choose the syntax that is easiest for your application.

Some stationary state-space models

▷ Example 1: An AR(1) model

Following Hamilton (1994b, 373–374), we can write the first-order autoregressive (AR(1)) model

$$y_t - \mu = \alpha(y_{t-1} - \mu) + \epsilon_t$$

as a state-space model with the observation equation

$$y_t = \mu + u_t$$

and the state equation

$$u_t = \alpha u_{t-1} + \epsilon_t$$

where the unobserved state is $u_t = y_t - \mu$.

Here we fit this model to data on the capacity utilization rate. The variable `lncaputil` contains data on the natural log of the capacity utilization rate for the manufacturing sector of the U.S. economy. We treat the series as first-difference stationary and fit its first-difference to an AR(1) process. Here we estimate the parameters of the above state-space form of the AR(1) model:

```
. use http://www.stata-press.com/data/r14/manufac
(St. Louis Fed (FRED) manufacturing data)
. constraint 1 [D.lncaputil]u = 1
. sspace (u L.u, state noconstant) (D.lncaputil u,  noerror), constraints(1)
searching for initial values ..........
(setting technique to bhhh)
Iteration 0:   log likelihood =  1515.8693
Iteration 1:   log likelihood =  1516.4187
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =    1516.44
Iteration 1:   log likelihood =    1516.44
State-space model
Sample: 1972m2 - 2008m12                        Number of obs      =       443
                                                Wald chi2(1)       =     61.73
Log likelihood =    1516.44                     Prob > chi2        =    0.0000
 ( 1)  [D.lncaputil]u = 1
```

lncaputil	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
u						
u L1.	.3523983	.0448539	7.86	0.000	.2644862	.4403104
D.lncaputil						
u	1	(constrained)				
_cons	-.0003558	.0005781	-0.62	0.538	-.001489	.0007773
var(u)	.0000622	4.18e-06	14.88	0.000	.000054	.0000704

```
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

The iteration log has three parts: the dots from the search for initial values, the log from finding the maximum, and the log from a refining step. Here is a description of the logic behind each part:

1. The quality of the initial values affect the speed and robustness of the optimization algorithm. `sspace` takes a few iterations in a nonlinear least-squares (NLS) algorithm to find good initial values and reports a dot for each (NLS) iteration.
2. This iteration log is the standard method by which Stata reports the search for the maximum likelihood estimates of the parameters in a nonlinear model.
3. Some of the parameters are transformed in the maximization process that `sspace` reports in part 2. After a maximum candidate is found in part 2, `sspace` looks for a maximum in the unconstrained space, checks that the Hessian of the log-likelihood function is of full rank, and reports these iterations as the refining step.

The header in the output describes the estimation sample, reports the log-likelihood function at the maximum, and gives the results of a Wald test against the null hypothesis that the coefficients on all the independent variables, state variables, and lagged state variables are zero. In this example, the null hypothesis that the coefficient on `L1.u` is zero is rejected at all conventional levels.

The estimation table reports results for the state equations, the observation equations, and the variance–covariance parameters. The estimated autoregressive coefficient of 0.3524 indicates that there is persistence in the first-differences of the log of the manufacturing rate. The estimated mean of the differenced series is -0.0004, which is smaller in magnitude than its standard error, indicating that there is no deterministic linear trend in the series.

Typing

```
. arima D.lncaputil, ar(1) technique(nr)
(output omitted)
```

produces nearly identical parameter estimates and standard errors for the mean and the autoregressive parameter. Because `sspace` estimates the variance of the state error while `arima` estimates the standard deviation, calculations are required to obtain the same results. The different parameterization of the variance parameter can cause small numerical differences.

◁

❑ Technical note

In some situations, the second part of the iteration log terminates but the refining step never converges. Only when the refining step converges does the maximization algorithm find interpretable estimates. If the refining step iterates without convergence, the parameters of the specified model are not identified by the data. (See Rothenberg [1971], Drukker and Wiggins [2004], and Davidson and MacKinnon [1993, sec. 5.2] for discussions of identification.)

❑

▷ Example 2: An ARMA(1,1) model

Following Harvey (1993, 95–96), we can write a zero-mean, first-order, autoregressive moving-average (ARMA(1,1)) model

$$y_t = \alpha y_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \tag{1}$$

as a state-space model with state equations

$$\begin{pmatrix} y_t \\ \theta\epsilon_t \end{pmatrix} = \begin{pmatrix} \alpha & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ \theta\epsilon_{t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ \theta \end{pmatrix} \epsilon_t \tag{2}$$

and observation equation

$$y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} y_t \\ \theta\epsilon_t \end{pmatrix} \tag{3}$$

The unobserved states in this model are $u_{1t} = y_t$ and $u_{2t} = \theta\epsilon_t$. We set the process mean to zero because economic theory and the previous example suggest that we should do so. Below we estimate the parameters in the state-space model by using the error-form syntax:

```
. constraint 2 [u1]L.u2 = 1
. constraint 3 [u1]e.u1 = 1
. constraint 4 [D.lncaputil]u1 = 1
. sspace (u1 L.u1 L.u2 e.u1, state noconstant) (u2 e.u1, state noconstant)
> (D.lncaputil u1, noconstant), constraints(2/4) covstate(diagonal)
searching for initial values ..........
(setting technique to bhhh)
Iteration 0:   log likelihood =  1478.5361
Iteration 1:   log likelihood =  1490.5202
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =   1531.255
Iteration 1:   log likelihood =   1531.255
State-space model
Sample: 1972m2 - 2008m12                        Number of obs      =       443
                                                Wald chi2(2)       =    333.84
Log likelihood =   1531.255                     Prob > chi2        =    0.0000
 ( 1)  [u1]L.u2 = 1
 ( 2)  [u1]e.u1 = 1
 ( 3)  [D.lncaputil]u1 = 1
```

lncaputil	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
u1						
u1 L1.	.8056815	.0522661	15.41	0.000	.7032418	.9081212
u2 L1.	1	(constrained)				
e.u1	1	(constrained)				
u2						
e.u1	-.5188453	.0701985	-7.39	0.000	-.6564317	-.3812588
D.lncaputil						
u1	1	(constrained)				
var(u1)	.0000582	3.91e-06	14.88	0.000	.0000505	.0000659

```
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

The command in the above output specifies two state equations, one observation equation, and two options. The first state equation defines u_{1t} and the second defines u_{2t} according to (2) above. The observation equation defines the process for `D.lncaputil` according to the one specified in (3) above. Several coefficients in (2) and (3) are set to 1, and constraints 2–4 place these restrictions on the model.

The estimated coefficient on `L.u1` in equation `u1`, 0.806, is the estimate of α in (2), which is the autoregressive coefficient in the ARMA model in (1). The estimated coefficient on `e.u1` in equation `u2`, -0.519, is the estimate of θ, which is the moving-average term in the ARMA model in (1).

This example highlights a difference between the error-form syntax and the covariance-form syntax. The error-form syntax used in this example includes only explicitly included errors. In contrast, the covariance-form syntax includes an error term in each equation, unless the `noerror` option is specified.

The default for `covstate()` also differs between the error-form syntax and the covariance-form syntax. Because the coefficients on the errors in the error-form syntax are frequently used to estimate the standard deviation of the errors, `covstate(identity)` is the default for the error-form syntax. In contrast, unit variances are less common in the covariance-form syntax, for which `covstate(diagonal)` is the default. In this example, we specified `covstate(diagonal)` to estimate a nonunitary variance for the state.

Typing

```
. arima D.lncaputil, noconstant ar(1) ma(1) technique(nr)
```
(*output omitted*)

produces nearly identical results. As in the AR(1) example above, `arima` estimates the standard deviation of the error term, while `sspace` estimates the variance. Although they are theoretically equivalent, the different parameterizations give rise to small numerical differences in the other parameters.

◁

▷ Example 3: A VAR(1) model

The variable `lnhours` contains data on the log of manufacturing hours, which we treat as first-difference stationary. We have a theory in which the process driving the changes in the log utilization rate affects the changes in the log of hours, but changes in the log hours do not affect changes in the log utilization rate. In line with this theory, we estimate the parameters of a lower triangular, first-order vector autoregressive (VAR(1)) process

$$\begin{pmatrix} \Delta\texttt{lncaputil}_t \\ \Delta\texttt{lnhours}_t \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 \\ \alpha_2 & \alpha_3 \end{pmatrix} \begin{pmatrix} \Delta\texttt{lncaputil}_{t-1} \\ \Delta\texttt{lnhours}_{t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} \tag{4}$$

where $\Delta y_t = y_t - y_{t-1}$, $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \epsilon_{2t})'$ and $\mathrm{Var}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}$. We can write this VAR(1) process as a state-space model with state equations

$$\begin{pmatrix} u_{1t} \\ u_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 \\ \alpha_2 & \alpha_3 \end{pmatrix} \begin{pmatrix} u_{1(t-1)} \\ u_{2(t-1)} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} \tag{5}$$

with $\mathrm{Var}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}$ and observation equations

$$\begin{pmatrix} \Delta\texttt{lncaputil} \\ \Delta\texttt{lnhours} \end{pmatrix} = \begin{pmatrix} u_{1t} \\ u_{2t} \end{pmatrix}$$

Below we estimate the parameters of the state-space model:

```
. constraint 5 [D.lncaputil]u1 = 1
. constraint 6 [D.lnhours]u2 = 1
. sspace (u1 L.u1, state noconstant)
>        (u2 L.u1 L.u2, state noconstant)
>        (D.lncaputil u1, noconstant noerror)
>        (D.lnhours u2, noconstant noerror),
>        constraints(5/6) covstate(unstructured)
searching for initial values ..........
(setting technique to bhhh)
Iteration 0:   log likelihood =  2789.6095
Iteration 1:   log likelihood =  2957.8299
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =  3211.7532
Iteration 1:   log likelihood =  3211.7532
State-space model
Sample: 1972m2 - 2008m12                        Number of obs      =       443
                                                Wald chi2(3)       =    166.87
Log likelihood =  3211.7532                     Prob > chi2        =    0.0000
 ( 1)  [D.lncaputil]u1 = 1
 ( 2)  [D.lnhours]u2 = 1
```

	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
u1						
u1 L1.	.353257	.0448456	7.88	0.000	.2653612	.4411528
u2						
u1 L1.	.1286218	.0394742	3.26	0.001	.0512537	.2059899
u2 L1.	-.3707083	.0434255	-8.54	0.000	-.4558208	-.2855959
D.lncaputil						
u1	1	(constrained)				
D.lnhours						
u2	1	(constrained)				
var(u1)	.0000623	4.19e-06	14.88	0.000	.0000541	.0000705
cov(u1,u2)	.000026	2.67e-06	9.75	0.000	.0000208	.0000312
var(u2)	.0000386	2.61e-06	14.76	0.000	.0000335	.0000437

```
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

Specifying `covstate(unstructured)` caused `sspace` to estimate the off-diagonal element of Σ. The output indicates that this parameter, `cov(u2,u1):_cons`, is small but statistically significant.

The estimated coefficient on `L.u1` in equation `u1`, 0.353, is the estimate of α_1 in (5). The estimated coefficient on `L.u1` in equation `u2`, 0.129, is the estimate of α_2 in (5). The estimated coefficient on `L.u1` in equation `u2`, -0.371, is the estimate of α_3 in (5).

For the VAR(1) model in (4), the estimated autoregressive coefficient for `D.lncaputil` is similar to the corresponding estimate in the univariate results in example 1. The estimated effect of `LD.lncaputil` on `D.lnhours` is 0.129, the estimated autoregressive coefficient of `D.lnhours` is -0.371, and both are statistically significant.

These estimates can be compared with those produced by typing

```
. constraint 101 [D_lncaputil]LD.lnhours = 0
. var D.lncaputil D.lnhours, lags(1) noconstant constraints(101)
(output omitted)
. matrix list e(Sigma)
(output omitted)
```

The `var` estimates are not the same as the `sspace` estimates because the generalized least-squares estimator implemented in `var` is only asymptotically equivalent to the ML estimator implemented in `sspace`, but the point estimates are similar. The comparison is useful for pedagogical purposes because the `var` estimator is relatively simple.

Some problems require constraining a covariance term to zero. If we wanted to constrain `cov(u2,u1):_cons` to zero, we could type

```
. constraint 7 [cov(u2,u1)]_cons = 0
. sspace (u1 L.u1, state noconstant)
>        (u2 L.u1 L.u2, state noconstant)
>        (D.lncaputil u1, noconstant noerror)
>        (D.lnhours u2, noconstant noerror),
>        constraints(5/7) covstate(unstructured)
(output omitted)
```

◁

▷ Example 4: A VARMA(1,1) model

We now extend the previous example by modeling `D.lncaputil` and `D.lnhours` as a first-order vector autoregressive moving-average (VARMA(1,1)) process. Building on the previous examples, we allow the lag of `D.lncaputil` to affect `D.lnhours` but we do not allow the lag of `D.lnhours` to affect the lag of `D.lncaputil`. Previous univariate analysis revealed that `D.lnhours` is better modeled as an autoregressive process than as an ARMA(1,1) process. As a result, we estimate the parameters of

$$\begin{pmatrix} \Delta\texttt{lncaputil}_t \\ \Delta\texttt{lnhours}_t \end{pmatrix} = \begin{pmatrix} \alpha_1 & 0 \\ \alpha_2 & \alpha_3 \end{pmatrix} \begin{pmatrix} \Delta\texttt{lncaputil}_{t-1} \\ \Delta\texttt{lnhours}_{t-1} \end{pmatrix} + \begin{pmatrix} \theta_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \epsilon_{1(t-1)} \\ \epsilon_{2(t-1)} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

We can write this VARMA(1,1) process as a state-space model with state equations

$$\begin{pmatrix} s_{1t} \\ s_{2t} \\ s_{3t} \end{pmatrix} = \begin{pmatrix} \alpha_1 & 1 & 0 \\ 0 & 0 & 0 \\ \alpha_2 & 0 & \alpha_3 \end{pmatrix} \begin{pmatrix} s_{1(t-1)} \\ s_{2(t-1)} \\ s_{3(t-1)} \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ \theta_1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

where the states are

$$\begin{pmatrix} s_{1t} \\ s_{2t} \\ s_{3t} \end{pmatrix} = \begin{pmatrix} \Delta\texttt{lncaputil}_t \\ \theta_1\epsilon_{1t} \\ \Delta\texttt{lnhours}_t \end{pmatrix}$$

and we simplify the problem by assuming that

$$\text{Var}\begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

Below we estimate the parameters of this model by using `sspace`:

```
. constraint 7  [u1]L.u2     = 1
. constraint 8  [u1]e.u1     = 1
. constraint 9  [u3]e.u3     = 1
. constraint 10  [D.lncaputil]u1 = 1
. constraint 11 [D.lnhours]u3 = 1
. sspace (u1 L.u1 L.u2 e.u1, state noconstant)
>        (u2 e.u1, state noconstant)
>        (u3 L.u1 L.u3 e.u3, state noconstant)
>        (D.lncaputil u1, noconstant)
>        (D.lnhours u3, noconstant),
>        constraints(7/11) technique(nr) covstate(diagonal)
searching for initial values ..........
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =  3156.0564
Iteration 1:   log likelihood =  3156.0564
State-space model
Sample: 1972m2 - 2008m12                        Number of obs      =       443
                                                Wald chi2(4)       =    427.55
Log likelihood =  3156.0564                     Prob > chi2        =    0.0000
 ( 1)  [u1]L.u2 = 1
 ( 2)  [u1]e.u1 = 1
 ( 3)  [u3]e.u3 = 1
 ( 4)  [D.lncaputil]u1 = 1
 ( 5)  [D.lnhours]u3 = 1
```

	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
u1						
u1 L1.	.8058031	.0522493	15.42	0.000	.7033964	.9082098
u2 L1.	1	(constrained)				
e.u1	1	(constrained)				
u2						
e.u1	-.518907	.0701848	-7.39	0.000	-.6564667	-.3813474
u3						
u1 L1.	.1734868	.0405156	4.28	0.000	.0940776	.252896
u3 L1.	-.4809376	.0498574	-9.65	0.000	-.5786563	-.3832188
e.u3	1	(constrained)				
D.lncaputil						
u1	1	(constrained)				
D.lnhours						
u3	1	(constrained)				
var(u1)	.0000582	3.91e-06	14.88	0.000	.0000505	.0000659
var(u3)	.0000382	2.56e-06	14.88	0.000	.0000331	.0000432

```
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

The estimates of the parameters in the model for `D.lncaputil` are similar to those in the univariate model fit in example 2. The estimates of the parameters in the model for `D.lnhours` indicate that the lag of `D.lncaputil` has a positive effect on `D.lnhours`.

◁

❑ Technical note

The `technique(nr)` option facilitates convergence in example 4. Fitting state-space models is notoriously difficult. Convergence problems are common. Four methods for overcoming convergence problems are 1) selecting an alternate optimization algorithm by using the `technique()` option, 2) using alternative starting values by specifying the `from()` option, 3) using starting values obtained by estimating the parameters of a restricted version of the model of interest, or 4) putting the variables on the same scale.

❑

▷ Example 5: A dynamic-factor model

Stock and Watson (1989, 1991) wrote a simple macroeconomic model as a dynamic-factor model, estimated the parameters by ML, and extracted an economic indicator. In this example, we estimate the parameters of a dynamic-factor model. In [TS] **sspace postestimation**, we extend this example and extract an economic indicator for the differenced series.

We have data on an industrial-production index, `ipman`; an aggregate weekly hours index, `hours`; and aggregate unemployment, `unemp`. `income` is real disposable income divided by 100. We rescaled real disposable income to avoid convergence problems.

We postulate a latent factor that follows an AR(2) process. Each measured variable is then related to the current value of that latent variable by a parameter. The state-space form of our model is

$$\begin{pmatrix} f_t \\ f_{t-1} \end{pmatrix} = \begin{pmatrix} \theta_1 & \theta_2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{t-1} \\ f_{t-2} \end{pmatrix} + \begin{pmatrix} \nu_t \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} \Delta\texttt{ipman}_t \\ \Delta\texttt{income}_t \\ \Delta\texttt{hours}_t \\ \Delta\texttt{unemp}_t \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} f_t + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix}$$

where

$$\text{Var}\begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}$$

The parameter estimates are

```
. use http://www.stata-press.com/data/r14/dfex
(St. Louis Fed (FRED) macro data)
. constraint 12 [lf]L.f = 1
. sspace (f L.f L.lf, state noconstant)
>        (lf L.f, state noconstant noerror)
>        (D.ipman f, noconstant)
>        (D.income f, noconstant)
>        (D.hours f, noconstant)
>        (D.unemp f, noconstant),
>        covstate(identity) constraints(12)
searching for initial values ..................
(setting technique to bhhh)
Iteration 0:   log likelihood = -674.18497
Iteration 1:   log likelihood = -667.23913
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood = -662.09507
Iteration 1:   log likelihood = -662.09507
State-space model
Sample: 1972m2 - 2008m11                        Number of obs      =       442
                                                Wald chi2(6)       =    751.95
Log likelihood = -662.09507                     Prob > chi2        =    0.0000
 ( 1)  [lf]L.f = 1
```

	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
f						
f L1.	.2651932	.0568663	4.66	0.000	.1537372	.3766491
lf L1.	.4820398	.0624635	7.72	0.000	.3596136	.604466
lf						
f L1.	1	(constrained)				
D.ipman f	.3502249	.0287389	12.19	0.000	.2938976	.4065522
D.income f	.0746338	.0217319	3.43	0.001	.0320401	.1172276
D.hours f	.2177469	.0186769	11.66	0.000	.1811407	.254353
D.unemp f	-.0676016	.0071022	-9.52	0.000	-.0815217	-.0536816
var(D.ipman)	.1383158	.0167086	8.28	0.000	.1055675	.1710641
var(D.income)	.2773808	.0188302	14.73	0.000	.2404743	.3142873
var(D.hours)	.0911446	.0080847	11.27	0.000	.0752988	.1069903
var(D.unemp)	.0237232	.0017932	13.23	0.000	.0202086	.0272378

```
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

The output indicates that the unobserved factor is quite persistent and that it is a significant predictor for each of the observed variables.

These models are frequently used to forecast the dependent variables and to estimate the unobserved factors. We present some illustrative examples in [TS] **sspace postestimation**. The `dfactor` command estimates the parameters of dynamic-factor models; see [TS] **dfactor**.

◁

Some nonstationary state-space models

▷ Example 6: A local-level model

Harvey (1989) advocates the use of STS models. These models parameterize the trends and seasonal components of a set of time series. The simplest STS model is the local-level model, which is given by

$$y_t = \mu_t + \epsilon_t$$

where

$$\mu_t = \mu_{t-1} + \nu_t$$

The model is called a local-level model because the level of the series is modeled as a random walk plus an idiosyncratic noise term. (The model is also known as the random-walk-plus-noise model.) The local-level model is nonstationary because of the random-walk component. When the variance of the idiosyncratic-disturbance ϵ_t is zero and the variance of the level-disturbance ν_t is not zero, the local-level model reduces to a random walk. When the variance of the level-disturbance ν_t is zero and the variance of the idiosyncratic-disturbance ϵ_t is not zero,

$$\mu_t = \mu_{t-1} = \mu$$

and the local-level model reduces to

$$y_t = \mu + \epsilon_t$$

which is a simple regression with a time-invariant mean. The parameter μ is not estimated in the state-space formulation below.

In this example, we fit weekly levels of the Standard and Poor's 500 Index to a local-level model. Because this model is already in state-space form, we fit `close` by typing

```
. use http://www.stata-press.com/data/r14/sp500w
. constraint 13 [z]L.z   = 1
. constraint 14 [close]z = 1
. sspace (z L.z, state noconstant) (close z, noconstant), constraints(13 14)
searching for initial values ..........
(setting technique to bhhh)
Iteration 0:   log likelihood =  -12582.89
Iteration 1:   log likelihood = -12577.146
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =  -12576.99
Iteration 1:   log likelihood =  -12576.99
State-space model
Sample: 1 - 3093                                Number of obs      =      3,093
Log likelihood =  -12576.99
 ( 1)  [z]L.z = 1
 ( 2)  [close]z = 1
```

close	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
z						
z L1.	1	(constrained)				
close						
z	1	(constrained)				
var(z)	170.3456	7.584909	22.46	0.000	155.4794	185.2117
var(close)	15.24858	3.392457	4.49	0.000	8.599486	21.89767

```
Note: Model is not stationary.
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

The results indicate that both components have nonzero variances. The output footer informs us that the model is nonstationary at the estimated parameter values.

◁

❑ Technical note

In the previous example, we estimated the parameters of a nonstationary state-space model. The model is nonstationary because one of the eigenvalues of the $\mathbf{A}$ matrix has unit modulus. That all the coefficients in the $\mathbf{A}$ matrix are fixed is also important. See Lütkepohl (2005, 636–637) for why the ML estimator for the parameters of a nonstationary state-model that is nonstationary because of eigenvalues with unit moduli from a fixed $\mathbf{A}$ matrix is still consistent and asymptotically normal.

❑

▷ Example 7: A local linear-trend model

In another basic STS model, known as the local linear-trend model, both the level and the slope of a linear time trend are random walks. Here are the state equations and the observation equation for a local linear-trend model for the level of industrial production contained in variable `ipman`:

$$\begin{pmatrix} \mu_t \\ \beta_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_{t-1} \\ \beta_{t-1} \end{pmatrix} + \begin{pmatrix} \nu_{1t} \\ \nu_{2t} \end{pmatrix}$$

$$\texttt{ipman}_t = \mu_t + \epsilon_t$$

The estimated parameters are

```
. use http://www.stata-press.com/data/r14/dfex
(St. Louis Fed (FRED) macro data)
. constraint 15 [f1]L.f1 = 1
. constraint 16 [f1]L.f2 = 1
. constraint 17 [f2]L.f2 = 1
. constraint 18 [ipman]f1   = 1
. sspace (f1 L.f1 L.f2, state noconstant)
>        (f2 L.f2, state noconstant)
>        (ipman f1, noconstant), constraints(15/18)
searching for initial values ..........
(setting technique to bhhh)
Iteration 0:   log likelihood = -362.93861
Iteration 1:   log likelihood = -362.12048
  (output omitted)
Refining estimates:
Iteration 0:   log likelihood =  -359.1266
Iteration 1:   log likelihood =  -359.1266
State-space model
Sample: 1972m1 - 2008m11                        Number of obs     =        443
Log likelihood =  -359.1266
 ( 1)  [f1]L.f1 = 1
 ( 2)  [f1]L.f2 = 1
 ( 3)  [f2]L.f2 = 1
 ( 4)  [ipman]f1 = 1
```

ipman	OIM Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
f1						
f1 L1.	1	(constrained)				
f2 L1.	1	(constrained)				
f2						
f2 L1.	1	(constrained)				
ipman						
f1	1	(constrained)				
var(f1)	.1473071	.0407156	3.62	0.000	.067506	.2271082
var(f2)	.0178752	.0065743	2.72	0.003	.0049898	.0307606
var(ipman)	.0354429	.0148186	2.39	0.008	.0063989	.0644868

```
Note: Model is not stationary.
Note: Tests of variances against zero are one sided, and the two-sided
      confidence intervals are truncated at zero.
```

There is little evidence that either of the variance parameters are zero. The fit obtained indicates that we could now proceed with specification testing and checks to see how well this model forecasts these data.

◁

Stored results

sspace stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_aux)	number of auxiliary parameters
e(k_eq)	number of equations in e(b)
e(k_dv)	number of dependent variables
e(k_obser)	number of observation equations
e(k_state)	number of state equations
e(k_obser_err)	number of observation-error terms
e(k_state_err)	number of state-error terms
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(chi2)	χ^2
e(p)	significance
e(tmin)	minimum time in sample
e(tmax)	maximum time in sample
e(stationary)	1 if the estimated parameters indicate a stationary model, 0 otherwise
e(rank)	rank of VCE
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	sspace
e(cmdline)	command as typed
e(depvar)	unoperated names of dependent variables in observation equations
e(obser_deps)	names of dependent variables in observation equations
e(state_deps)	names of dependent variables in state equations
e(covariates)	list of covariates
e(indeps)	independent variables
e(tvar)	variable denoting time within groups
e(eqnames)	names of equations
e(title)	title in estimation output
e(tmins)	formatted minimum time
e(tmaxs)	formatted maximum time
e(R_structure)	structure of observed-variable-error covariance matrix
e(Q_structure)	structure of state-error covariance matrix
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(opt)	type of optimization
e(method)	likelihood method
e(initial_values)	type of initial values
e(technique)	maximization technique
e(tech_steps)	iterations taken in maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	parameter vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(gamma)	mapping from parameter vector to state-space matrices
e(A)	estimated A matrix
e(B)	estimated B matrix
e(C)	estimated C matrix
e(D)	estimated D matrix
e(F)	estimated F matrix
e(G)	estimated G matrix
e(chol_R)	Cholesky factor of estimated R matrix
e(chol_Q)	Cholesky factor of estimated Q matrix
e(chol_Sz0)	Cholesky factor of initial state covariance matrix
e(z0)	initial state vector augmented with a matrix identifying nonstationary components
e(d)	additional term in diffuse initial state vector, if nonstationary model
e(T)	inner part of quadratic form for initial state covariance in a partially nonstationary model
e(M)	outer part of quadratic form for initial state covariance in a partially nonstationary model
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

Methods and formulas

Recall that our notation for linear state-space models with time-invariant coefficient matrices is

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{B}\mathbf{x}_t + \mathbf{C}\boldsymbol{\epsilon}_t$$
$$\mathbf{y}_t = \mathbf{D}\mathbf{z}_t + \mathbf{F}\mathbf{w}_t + \mathbf{G}\boldsymbol{\nu}_t$$

where

$\mathbf{z}_t$ is an $m \times 1$ vector of unobserved state variables;

$\mathbf{x}_t$ is a $k_x \times 1$ vector of exogenous variables;

$\boldsymbol{\epsilon}_t$ is a $q \times 1$ vector of state-error terms, $(q \leq m)$;

$\mathbf{y}_t$ is an $n \times 1$ vector of observed endogenous variables;

$\mathbf{w}_t$ is a $k_w \times 1$ vector of exogenous variables;

$\boldsymbol{\nu}_t$ is an $r \times 1$ vector of observation-error terms, $(r \leq n)$; and

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{F}$, and $\mathbf{G}$ are parameter matrices.

The equations for $\mathbf{z}_t$ are known as the state equations, and the equations for $\mathbf{y}_t$ are known as the observation equations.

The error terms are assumed to be zero mean, normally distributed, serially uncorrelated, and uncorrelated with each other;

$$\boldsymbol{\epsilon}_t \sim N(0, \mathbf{Q})$$
$$\boldsymbol{\nu}_t \sim N(0, \mathbf{R})$$
$$E[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_s'] = \mathbf{0} \text{ for all } s \neq t$$
$$E[\boldsymbol{\epsilon}_t \boldsymbol{\nu}_s'] = \mathbf{0} \text{ for all } s \text{ and } t$$

sspace estimates the parameters of linear state-space models by maximum likelihood. The Kalman filter is a method for recursively obtaining linear, least-squares forecasts of $\mathbf{y}_t$ conditional on past information. These forecasts are used to construct the log likelihood, assuming normality and stationarity. When the model is nonstationary, a diffuse Kalman filter is used.

Hamilton (1994a; 1994b, 389) shows that the QML estimator, obtained when the normality assumption is dropped, is consistent and asymptotically normal, although the variance–covariance matrix of the estimator (VCE) must be estimated by the Huber/White/sandwich estimator. Hamilton's discussion applies to stationary models, and specifying vce(robust) produces a consistent estimator of the VCE when the errors are not normal.

Methods for computing the log likelihood differ in how they calculate initial values for the Kalman filter when the model is stationary, how they compute a diffuse Kalman filter when the model is nonstationary, and whether terms for initial states are included. sspace offers the method(hybrid), method(dejong), and method(kdiffuse) options for computing the log likelihood. All three methods handle both stationary and nonstationary models.

method(hybrid), the default, uses the initial values for the states implied by stationarity to initialize the Kalman filter when the model is stationary. Hamilton (1994b, 378) discusses this method of computing initial values for the states and derives a log-likelihood function that does not include terms for the initial states. When the model is nonstationary, method(hybrid) uses the De Jong (1988, 1991) diffuse Kalman filter and log-likelihood function, which includes terms for the initial states.

method(dejong) uses the stationary De Jong (1988) method when the model is stationary and the De Jong (1988, 1991) diffuse Kalman filter when the model is nonstationary. The stationary De Jong (1988) method estimates initial values for the Kalman filter as part of the log-likelihood computation, as in De Jong (1988).

method(kdiffuse) implements the seldom-used large-κ diffuse approximation to the diffuse Kalman filter when the model is nonstationary and uses initial values for the states implied by stationarity when the model is stationary. The log likelihood does not include terms for the initial states in either case. We recommend that you do not use method(kdiffuse) except to replicate older results computed using this method.

De Jong (1988, 1991) and De Jong and Chu-Chun-Lin (1994) derive the log likelihood and a diffuse Kalman filter for handling nonstationary data. De Jong (1988) replaces the stationarity assumption with a time-immemorial assumption, which he uses to derive the log-likelihood function, an initial state vector, and a covariance of the initial state vector when the model is nonstationary. By default, and when method(hybrid) or method(dejong) is specified, sspace uses the diffuse Kalman filter given in definition 5 of De Jong and Chu-Chun-Lin (1994). This method uses theorem 3 of De Jong and Chu-Chun-Lin (1994) to compute the covariance of the initial states. When using this method, sspace saves the matrices from their theorem 3 in e(), although the names are changed. e(Z) is their $\mathbf{U}_1$, e(T) is their $\mathbf{U}_2$, e(A) is their $\mathbf{T}$, and e(M) is their $\mathbf{M}$.

See De Jong (1988, 1991) and De Jong and Chu-Chun-Lin (1994) for the details of the De Jong diffuse Kalman filter.

Practical estimation and inference require that the maximum likelihood estimator be consistent and normally distributed in large samples. These statistical properties of the maximum likelihood estimator are well established when the model is stationary; see Caines (1988, chap. 5 and 7), Hamilton (1994b, 388–389), and Hannan and Deistler (1988, chap. 4). When the model is nonstationary, additional assumptions must hold for the maximum likelihood estimator to be consistent and asymptotically normal; see Harvey (1989, sec. 3.4), Lütkepohl (2005, 636–637), and Schneider (1988). Chang, Miller, and Park (2009) show that the ML and the QML estimators are consistent and asymptotically normal for a class of nonstationary state-space models.

We now give an intuitive version of the Kalman filter. `sspace` uses theoretically equivalent, but numerically more stable, methods. For each time t, the Kalman filter produces the conditional expected state vector $\mathbf{z}_{t|t}$ and the conditional covariance matrix $\mathbf{\Omega}_{t|t}$; both are conditional on information up to and including time t. Using the model and previous period results, for each t we begin with

$$\begin{aligned}\mathbf{z}_{t|t-1} &= \mathbf{A}\mathbf{z}_{t-1|t-1} + \mathbf{B}\mathbf{x}_t \\ \mathbf{\Omega}_{t|t-1} &= \mathbf{A}\mathbf{\Omega}_{t-1|t-1}\mathbf{A}' + \mathbf{C}\mathbf{Q}\mathbf{C}' \\ \mathbf{y}_{t|t-1} &= \mathbf{D}\mathbf{z}_{t|t-1} + \mathbf{F}\mathbf{w}_t\end{aligned} \tag{6}$$

The residuals and the mean squared error (MSE) matrix of the forecast error are

$$\begin{aligned}\widetilde{\boldsymbol{\nu}}_{t|t} &= \mathbf{y}_t - \mathbf{y}_{t|t-1} \\ \mathbf{\Sigma}_{t|t} &= \mathbf{D}\mathbf{\Omega}_{t|t-1}\mathbf{D}' + \mathbf{G}\mathbf{R}\mathbf{G}'\end{aligned} \tag{7}$$

In the last steps, we update the conditional expected state vector and the conditional covariance with the time t information:

$$\begin{aligned}\mathbf{z}_{t|t} &= \mathbf{z}_{t|t-1} + \mathbf{\Omega}_{t|t-1}\mathbf{D}\mathbf{\Sigma}_{t|t}^{-1}\widetilde{\boldsymbol{\nu}}_{t|t} \\ \mathbf{\Omega}_{t|t} &= \mathbf{\Omega}_{t|t-1} - \mathbf{\Omega}_{t|t-1}\mathbf{D}\mathbf{\Sigma}_{t|t}^{-1}\mathbf{D}'\mathbf{\Omega}_{t|t-1}\end{aligned} \tag{8}$$

Equations (6)–(8) are the Kalman filter. The equations denoted by (6) are the one-step predictions. The one-step predictions do not use contemporaneous values of $\mathbf{y}_t$; only past values of $\mathbf{y}_t$, past values of the exogenous $\mathbf{x}_t$, and contemporaneous values of $\mathbf{x}_t$ are used. Equations (7) and (8) form the update step of the Kalman filter; they incorporate the contemporaneous dependent variable information into the predicted states.

The Kalman filter requires initial values for the states and a covariance matrix for the initial states to start off the recursive process. Hamilton (1994b) discusses how to compute initial values for the Kalman filter assuming stationarity. This method is used by default when the model is stationary. De Jong (1988) discusses how to estimate initial values by maximum likelihood; this method is used when `method(dejong)` is specified.

Letting $\boldsymbol{\delta}$ be the vector of parameters in the model, Lütkepohl (2005) and Harvey (1989) show that the log-likelihood function for the parameters of a stationary model is given by

$$\ln L(\boldsymbol{\delta}) = -0.5\left\{nT\ln(2\pi) + \sum_{t=1}^{T}\ln(|\mathbf{\Sigma}_{t|t-1}|) + \sum_{t=1}^{T}\mathbf{e}_t{}'\mathbf{\Sigma}_{t|t-1}^{-1}\mathbf{e}_t\right\}$$

where $\mathbf{e}_t = (\mathbf{y}_t - \mathbf{y}_{t|t-1})$ depends on $\boldsymbol{\delta}$ and $\mathbf{\Sigma}$ also depends on $\boldsymbol{\delta}$.

The variance–covariance matrix of the estimator (VCE) is estimated by the observed information matrix (OIM) estimator by default. Specifying `vce(robust)` causes `sspace` to use the Huber/White/sandwich estimator. Both estimators of the VCE are standard and documented in Hamilton (1994b).

Hamilton (1994b), Hannan and Deistler (1988), and Caines (1988) show that the ML estimator is consistent and asymptotically normal when the model is stationary. Schneider (1988) establishes consistency and asymptotic normality when the model is nonstationary because $\mathbf{A}$ has some eigenvalues with modulus 1 and there are no unknown parameters in $\mathbf{A}$.

Not all state-space models are identified, as discussed in Hamilton (1994b) and Lütkepohl (2005). `sspace` checks for local identification at the optimum. `sspace` will not declare convergence unless the Hessian is full rank. This check for local identifiability is due to Rothenberg (1971).

Specifying `method(dejong)` causes `sspace` to maximize the log-likelihood function given in section 2 (vii) of De Jong (1988). This log-likelihood function includes the initial states as parameters to be estimated. We use some of the methods in Casals, Sotoca, and Jerez (1999) for computing the De Jong (1988) log-likelihood function.

References

Anderson, B. D. O., and J. B. Moore. 1979. *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall.

Brockwell, P. J., and R. A. Davis. 1991. *Time Series: Theory and Methods*. 2nd ed. New York: Springer.

Caines, P. E. 1988. *Linear Stochastic Systems*. New York: Wiley.

Casals, J., S. Sotoca, and M. Jerez. 1999. A fast and stable method to compute the likelihood of time invariant state-space models. *Economics Letters* 65: 329–337.

Chang, Y., J. I. Miller, and J. Y. Park. 2009. Extracting a common stochastic trend: Theory with some applications. *Journal of Econometrics* 150: 231–247.

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.

De Jong, P. 1988. The likelihood for a state space model. *Biometrika* 75: 165–169.

——. 1991. The diffuse Kalman filter. *Annals of Statistics* 19: 1073–1083.

De Jong, P., and S. Chu-Chun-Lin. 1994. Stationary and non-stationary state space models. *Journal of Time Series Analysis* 15: 151–166.

Drukker, D. M., and V. L. Wiggins. 2004. Verifying the solution from a nonlinear solver: A case study: Comment. *American Economic Review* 94: 397–399.

Hamilton, J. D. 1994a. State-space models. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 3039–3080. Amsterdam: Elsevier.

——. 1994b. *Time Series Analysis*. Princeton: Princeton University Press.

Hannan, E. J., and M. Deistler. 1988. *The Statistical Theory of Linear Systems*. New York: Wiley.

Harvey, A. C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.

——. 1993. *Time Series Models*. 2nd ed. Cambridge, MA: MIT Press.

Lütkepohl, H. 2005. *New Introduction to Multiple Time Series Analysis*. New York: Springer.

Rothenberg, T. J. 1971. Identification in parametric models. *Econometrica* 39: 577–591.

Schneider, W. 1988. Analytical uses of Kalman filtering in econometrics: A survey. *Statistical Papers* 29: 3–33.

Stock, J. H., and M. W. Watson. 1989. New indexes of coincident and leading economic indicators. In *NBER Macroeconomics Annual 1989*, ed. O. J. Blanchard and S. Fischer, vol. 4, 351–394. Cambridge, MA: MIT Press.

——. 1991. A probability model of the coincident economic indicators. In *Leading Economic Indicators: New Approaches and Forecasting Records*, ed. K. Lahiri and G. H. Moore, 63–89. Cambridge: Cambridge University Press.

Also see

[TS] **sspace postestimation** — Postestimation tools for sspace

[TS] **arima** — ARIMA, ARMAX, and other dynamic regression models

[TS] **dfactor** — Dynamic-factor models

[TS] **tsset** — Declare data to be time-series data

[TS] **ucm** — Unobserved-components model

[TS] **var** — Vector autoregressive models

[U] **20 Estimation and postestimation commands**