Title

stata.com

mgarch dvech postestimation — Postestimation tools for mgarch dvech

Description

The following standard postestimation commands are available after `mgarch dvech`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`forecast`	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

Syntax for predict

```
predict [type] {stub* | newvarlist} [if] [in] [, statistic options]
```

statistic	Description
Main	
`xb`	linear prediction; the default
`residuals`	residuals
`variance`	conditional variances and covariances

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

options	Description
Options	
`equation(`*eqnames*`)`	names of equations for which predictions are made
`dynamic(`*time_constant*`)`	begin dynamic forecast at specified time

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`xb`, the default, calculates the linear predictions of the dependent variables.

`residuals` calculates the residuals.

`variance` predicts the conditional variances and conditional covariances.

Options

`equation(`*eqnames*`)` specifies the equation for which the predictions are calculated. Use this option to predict a statistic for a particular equation. Equation names, such as `equation(income)`, are used to identify equations.

One equation name may be specified when predicting the dependent variable, the residuals, or the conditional variance. For example, specifying `equation(income)` causes `predict` to predict `income`, and specifying `variance equation(income)` causes predict to predict the conditional variance of income.

Two equations may be specified when predicting a conditional variance or covariance. For example, specifying `equation(income, consumption) variance` causes `predict` to predict the conditional covariance of `income` and `consumption`.

`dynamic(`*time_constant*`)` specifies when `predict` starts producing dynamic forecasts. The specified *time_constant* must be in the scale of the time variable specified in `tsset`, and the *time_constant* must be inside a sample for which observations on the dependent variables are available. For example, `dynamic(tq(2008q4))` causes dynamic predictions to begin in the fourth quarter of 2008, assuming that your time variable is quarterly; see [D] **datetime**. If the model contains exogenous variables, they must be present for the whole predicted sample. `dynamic()` may not be specified with `residuals`.

Remarks and examples

stata.com

We assume that you have already read [TS] **mgarch dvech**. In this entry, we illustrate some of the features of `predict` after using `mgarch dvech` to estimate the parameters of diagonal vech MGARCH models.

▷ Example 1: Dynamic forecasts

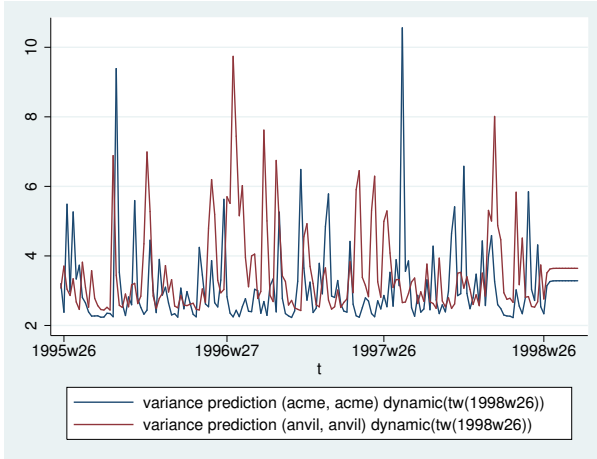
In this example, we obtain dynamic predictions for the Acme Inc. and Anvil Inc. fictional widget data modeled in example 3 of [TS] **mgarch dvech**. We begin by reestimating the parameters of the model.

```
. use http://www.stata-press.com/data/r13/acme

. constraint 1 [L.ARCH]1_1 = [L.ARCH]2_2

. constraint 2 [L.GARCH]1_1 = [L.GARCH]2_2

. mgarch dvech (acme = L.acme) (anvil = L.anvil), arch(1) garch(1)
> constraints(1 2)

Getting starting values
(setting technique to bhhh)
Iteration 0:   log likelihood = -6087.0665  (not concave)
Iteration 1:   log likelihood = -6022.2046
Iteration 2:   log likelihood = -5986.6152
Iteration 3:   log likelihood = -5976.5739
Iteration 4:   log likelihood = -5974.4342
Iteration 5:   log likelihood = -5974.4046
Iteration 6:   log likelihood = -5974.4036
Iteration 7:   log likelihood = -5974.4035

Estimating parameters
(setting technique to bhhh)
Iteration 0:   log likelihood = -5974.4035
Iteration 1:   log likelihood =  -5973.812
Iteration 2:   log likelihood = -5973.8004
Iteration 3:   log likelihood = -5973.7999
Iteration 4:   log likelihood = -5973.7999

Diagonal vech MGARCH model

Sample: 1969w35 - 1998w25                       Number of obs   =      1499
Distribution: Gaussian                          Wald chi2(2)    =    272.47
Log likelihood =   -5973.8                      Prob > chi2     =    0.0000

 ( 1)  [L.ARCH]1_1 - [L.ARCH]2_2 = 0
 ( 2)  [L.GARCH]1_1 - [L.GARCH]2_2 = 0
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
acme						
acme						
L1.	.3365278	.0255134	13.19	0.000	.2865225	.3865331
_cons	1.124611	.060085	18.72	0.000	1.006847	1.242376
anvil						
anvil						
L1.	.3151955	.0263287	11.97	0.000	.2635922	.3667988
_cons	1.215786	.0642052	18.94	0.000	1.089947	1.341626
Sigma0						
1_1	1.889237	.2168733	8.71	0.000	1.464173	2.314301
2_1	.4599576	.1139843	4.04	0.000	.2365525	.6833626
2_2	2.063113	.2454633	8.40	0.000	1.582014	2.544213
L.ARCH						
1_1	.2813443	.0299124	9.41	0.000	.222717	.3399716
2_1	.181877	.0335393	5.42	0.000	.1161412	.2476128
2_2	.2813443	.0299124	9.41	0.000	.222717	.3399716
L.GARCH						
1_1	.1487581	.0697531	2.13	0.033	.0120445	.2854716
2_1	.085404	.1446524	0.59	0.555	-.1981094	.3689175
2_2	.1487581	.0697531	2.13	0.033	.0120445	.2854716

Now we use `tsappend` (see [TS] **tsappend**) to extend the data, use `predict` to obtain the dynamic predictions, and graph the predictions.

```
. tsappend, add(12)
. predict H*, variance dynamic(tw(1998w26))
. tsline H_acme_acme H_anvil_anvil if t>=tw(1995w25), legend(rows(2))
```



The graph shows that the in-sample predictions are similar for the conditional variances of Acme Inc. and Anvil Inc. and that the dynamic forecasts converge to similar levels. It also shows that the ARCH and GARCH parameters cause substantial time-varying volatility. The predicted conditional variance of `acme` ranges from lows of just over 2 to highs above 10.

◁

▷ Example 2: Predicting in-sample conditional variances

In this example, we obtain the in-sample predicted conditional variances of the returns for the fictional Acme Inc., which we modeled in example 4 of [TS] **mgarch dvech**. First, we reestimate the parameters of the model.

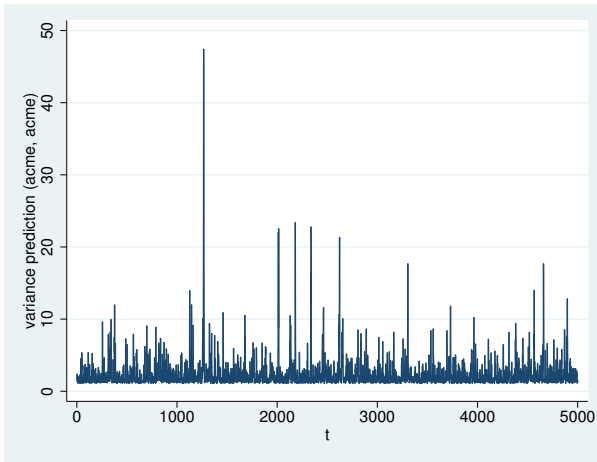
```
. use http://www.stata-press.com/data/r13/aacmer, clear
. mgarch dvech (acme anvil = , noconstant), arch(1/2) garch(1)
Getting starting values
(setting technique to bhhh)
Iteration 0:   log likelihood = -18417.243  (not concave)
Iteration 1:   log likelihood = -18215.005
Iteration 2:   log likelihood = -18199.691
Iteration 3:   log likelihood = -18136.699
Iteration 4:   log likelihood = -18084.256
Iteration 5:   log likelihood = -17993.662
Iteration 6:   log likelihood =   -17731.1
Iteration 7:   log likelihood = -17629.505
(switching technique to nr)
Iteration 8:   log likelihood = -17548.172
Iteration 9:   log likelihood = -17544.987
Iteration 10:  log likelihood = -17544.937
Iteration 11:  log likelihood = -17544.937
```

```
Estimating parameters
(setting technique to bhhh)
Iteration 0:   log likelihood = -17544.937
Iteration 1:   log likelihood = -17544.937
Diagonal vech MGARCH model
Sample: 1 - 5000                                    Number of obs   =      5000
Distribution: Gaussian                              Wald chi2(.)    =         .
Log likelihood = -17544.94                          Prob > chi2     =         .
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Sigma0						
1_1	1.026283	.0823348	12.46	0.000	.8649096	1.187656
2_1	.4300997	.0590294	7.29	0.000	.3144042	.5457952
2_2	1.019753	.0837146	12.18	0.000	.8556751	1.18383
L.ARCH						
1_1	.2878739	.02157	13.35	0.000	.2455975	.3301504
2_1	.1036685	.0161446	6.42	0.000	.0720256	.1353114
2_2	.2034196	.019855	10.25	0.000	.1645044	.2423347
L2.ARCH						
1_1	.1837825	.0274555	6.69	0.000	.1299706	.2375943
2_1	.0884425	.02208	4.01	0.000	.0451665	.1317185
2_2	.2025718	.0272639	7.43	0.000	.1491355	.256008
L.GARCH						
1_1	.0782467	.053944	1.45	0.147	-.0274816	.183975
2_1	.2888104	.0818303	3.53	0.000	.1284261	.4491948
2_2	.201618	.0470584	4.28	0.000	.1093853	.2938508

Now we use `predict` to obtain the in-sample conditional variances of `acme` and use `tsline` (see [TS] **tsline**) to graph the results.

```
. predict h_acme, variance eq(acme, acme)
. tsline h_acme
```



The graph shows that the predicted conditional variances vary substantially over time, as the parameter estimates indicated.

Because there are no covariates in the model for `acme`, specifying `xb` puts a prediction of 0 in each observation, and specifying `residuals` puts the value of the dependent variable into the prediction. ◁

Methods and formulas

All one-step predictions are obtained by substituting the parameter estimates into the model. The estimated unconditional variance matrix of the disturbances, $\widehat{\boldsymbol{\Sigma}}$, is the initial value for the ARCH and GARCH terms. The postestimation routines recompute $\widehat{\boldsymbol{\Sigma}}$ using the prediction sample, the parameter estimates stored in `e(b)`, and (4) in *Methods and formulas* of [TS] **mgarch dvech**.

For observations in which the residuals are missing, the estimated unconditional variance matrix of the disturbances is used in place of the outer product of the residuals.

Dynamic predictions of the dependent variables use previously predicted values beginning in the period specified by `dynamic()`.

Dynamic variance predictions are implemented by substituting $\widehat{\boldsymbol{\Sigma}}$ for the outer product of the residuals beginning in the period specified by `dynamic()`.

Also see

[TS] **mgarch dvech** — Diagonal vech multivariate GARCH models

[U] 20 Estimation and postestimation commands