Estimating ARDL models in R

Notes on the use of the `R` package `ardl` v.0.0.4

Fernando C. Barbi[*]

June 07, 2016

Abstract

The ‘ardl‘ (Auto Regressive Distributed Lag) package estimates time series dynamic models with lagged dependent variables and lagged regressors. It is specially useful to study time relations when the structure of the models are not imposed *a priori* by theory. The flexibility offered by a variable number of lags and the possibilty to model in levels lead to models highly adjusted to data. The current version of this package allows for unrestricted estimations of constant parameter models for short and long-term relations. Future versions will allow for restricted estimations (see section *Future Developments* for other features under work).

Introduction

An univariate model (single equation) is estimated with regressors in levels using the ARDL framework presented in M. H. Pesaran and Shin (1999) regardless of whether the regressors are stationary, I(0), or have a unit root, I(1). A nice introduction to the subject is Hassler and Wolters (2005).

This package relies on `dynlm` for estimation so only *unrestricted coefficient models* can be estimated, the advantage is that this routine is based on the robust `QR` decomposition that behaves well even in case of strong collinearity among regressors.

The package supports the automatic identification of the best model according to different selection criteria (BIC, AIC, R2 and LL). It also provides tools to visualize the cointegration (long-term) relation and to test it using the bounds testing procedure presented in M. Pesaran, Shin, and Smith (2001).

Some ARDL theory

Assume the model for y_t has no trend and $\mathbf{x_t}$ is a set of *weakly exogenous* K regressors

$$y_t = \alpha + \phi y_{t-1} + \beta' \mathbf{x}_t + \psi d_t + u_t$$

where d_t stands for the the intervention variables (dummies) that are not to be lagged.

We also assume some components of $\mathbf{x}_t$ to be unit-root, I(1), processes as in

[*] fcbarbi@gmail.com

$$x_t = x_{t-1} + \varepsilon_t$$

The regressors $\mathbf{x}_t$ may be I(0) or I(1) so we start by assuming the existence of a long-term (**cointegrating**) relation between I(1) regressors so that by combining these terms we obtain a stationary, I(0), term.

M. H. Pesaran and Shin (1999) show that the least square (LS) estimation of this model provides consistent estimators with *super-consistence* properties for the long-term coefficients: these estimators converge to the true parameter values at speed proportional to T, faster than the usual $\sqrt{T}$ convergence of LS estimators. This is particularly interesting when working with small samples. The long-term coefficients $\hat{\Theta}$ are a function of the estimated short-term coefficients $\hat{\beta}$ and $\hat{\phi}$ as given by

$$\hat{\Theta} = g(\hat{\beta}, \hat{\phi}) = \frac{\hat{\beta}}{1 - \hat{\phi}} = g(\hat{\Psi})$$

The variance of Θ can be approximated by the delta method

$$V(\hat{\Theta}) = \left(\frac{\partial g(\hat{\Psi})}{\partial \hat{\Psi}} \right)' V(\hat{\Psi}) \left(\frac{\partial g(\hat{\Psi})}{\partial \hat{\Psi}} \right)$$

After some algebra we get the result used in the code to get the variance of each of the core components of the specification

$$V(\hat{\Theta}) = \frac{\hat{\sigma}_u^2}{(1 - \hat{\phi})^2}(1, \hat{\Theta}) \frac{1}{D_T} \left[\begin{array}{cc} \sum(y_{t-1} - \overline{y})^2 & -\sum(y_{t-1} - \overline{y})(x_t - \overline{x}) \\ -\sum(y_{t-1} - \overline{y})(x_t - \overline{x}) & \sum(x_t - \overline{x})^2 \end{array} \right] \left(\begin{array}{c} 1 \\ \hat{\Theta} \end{array} \right)$$

This result is eq.2.20 in M. Pesaran, Shin, and Smith (2001). The term $\overline{x}$ is the sample mean and D_T is defined as

$$D_T = \left[\sum (x_t - \overline{x})^2 \right] \left[\sum (y_{t-1} - \overline{y})^2 \right] - \left[\sum (y_{t-1} - \overline{y})(x_t - \overline{x}) \right]^2$$

The cointegration term is stationary, $(y_t - \hat{\Theta} x_t) \sim I(0)$. Once it is calculated you can check the significance of each individual estimator with a t-test. The results should be confirmed by a Wald test, presented in M. Pesaran, Shin, and Smith (2001) as the "bounds test", to compare two specifications, one with the regressors in levels and the other without them. In the model

$$d(y_t) = \alpha + \sum_{j=0}^{m} \pi_j \left(\begin{array}{c} y_{t-1-j} \\ x_{t-j} \end{array} \right) + \sum_{i=0}^{p} \phi_i \left(\begin{array}{c} d(y_{t-i}) \\ d(x_{t-i}) \end{array} \right) + \varepsilon_t$$

where $d()$ is the first difference operator, the null hypothesis of the bounds test is that $\pi_0 = \pi_1 = \cdots = \pi_m = 0$, $\pi_t = (\pi_t^y, \pi_t^x)$ and $\phi_t = (\phi_t^y, \phi_t^x)$.

Package Overview

This package has essentially 4 functions in addition to those usually available to `lm` models:

1. `ardl()` is the core function that relies on package `dynlm` to estimate the dynamic models. It can be called with a `quiet=TRUE` option to operate in silence so it can be called in other tools.
2. `auto.ardl()` uses `ardl()` to find the best specification. It can be called with `verbose=TRUE` to show all the models under test.
3. `coint()` prints the two sets of coefficients: long-run (LR) and short-run (SR). It can generate output directly to files in `.txt` or `.tex` formats.
4. `bounds.test()` tests the existence of a long-run relationship in models with I(0) or I(1) regressors using M. Pesaran, Shin, and Smith (2001) critical values.

Finally, note that `print()` and `summary()` work as for any linear model (`lm`).

Some Conventions

The functions `ardl()` and `auto.ardl()` receive the "canonical" equation in the form `y ~ x1+x2|x3` that means that `y` depends on a variable number of lags of `x1` and `x2` while `x3` must be taken as is, this term is generallay a dummy so it should not be differenced or lagged, hence its name "fixed". Note the `|` character that is used to divide the terms, you can certainly have more than one fixed term as in `y ~ x1+x2|x3+x4`.

Assuming `case=5` the model is estimated with an unrestricted intercept and an unrestricted trend with lags for `y = 1` and `x = c(1,2)` the "expanded" equation is therefore `y ~ +1+trend(y) + L(y,1) + x1+L(x1,1)+x2+L(x2,1)+L(x2,2) + x3`. This format is convenient for calling the `dynlm()` to do the actual estimation.

The case number informs on the existence of an intercept and a trend in the model following the convention of M. Pesaran, Shin, and Smith (2001):

Case Number	Description
1	no intercept, no trend
2	restricted intercert and no trend (not supported)
3	unrestricted intercert and no trend
4	unrestricted intercept and restricted trend (not supported)
5	unrestricted intercept and unrestricted trend

Examples

```r
#devtools::install_github("fcbarbi/ardl")
require(ardl)
```

```
## Loading required package: ardl
```

```
data(br_month)
```

An ARDL(2,1,1) model structure for the monetary policy rate `mpr` with two regressors: prices `cpi` and the exchange rate `reer` with at most one lag each as in

$$i_t = \alpha + \phi_1 i_{t-1} + \phi_2 i_{t-2} + \beta_1 \pi_t + \beta_2 \pi_{t-1} + \beta_3 s_t + \beta_4 s_{t-1} + \varepsilon_t$$

where i_t is the interest rate, π_t is inflation and s_t is the exchange rate.

Function: ardl()

This model is estimated with monthly data from Brazil with the command

```
m1 <- ardl( mpr~cpi+reer | d_lula, data=br_month,
            ylag=2, xlag=c(1,1), case=3 )
```

```
##
## Dataset adjustment to the common sample of all regressors:
## Original dataset from  2001(1) to 2015(2)
## Adjusted dataset from  2001(1) to 2015(1)
##
## AutoRegressive Distributed Lag model
##
## Time series regression with "zooreg" data:
## Start = 2001(3), End = 2015(1)
##
## Call:
## dynlm::dynlm(formula = formula(fm), data = data, subset = subset)
##
## Coefficients:
## (Intercept)    L(mpr, 1)    L(mpr, 2)          cpi    L(cpi, 1)
##   -0.402581     1.700397    -0.733307     0.069484    -0.035500
##        reer   L(reer, 1)       d_lula
##    0.008958    -0.002349    -0.541249
##
##
##  Long-term coefficients:
##         cpi        reer      d_lula
##   1.0326097   0.2008177 -16.4460621
##
##
##  Short-term coefficients
##  (Intercept)    L(d(mpr))       d(cpi)      d(reer)       d_lula
## -0.406626972  0.712596057  0.035799573  0.008433137 -0.059325894
##     L(coint)
## -0.033565499
```

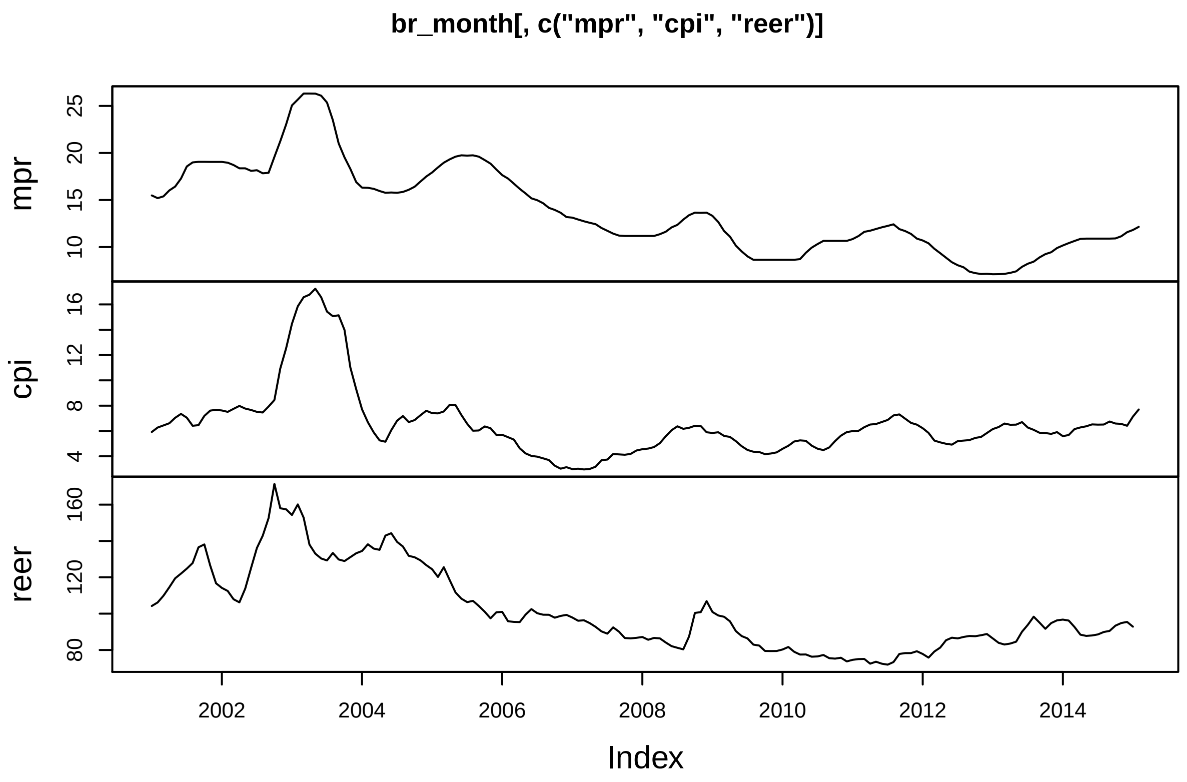
Note that `ardl()` tests for the existence of `NA` in data and automatically adjusts the top and bottom of the dataset. You can check this by including `prod` in the model, this data is only available from January 2003 up to December 2014:

```
m1 <- ardl( mpr~cpi+prod+reer | d_lula, data=br_month,
            ylag=2, xlag=c(1,1,1), case=3 )

Dataset adjustment to the common sample of all regressors:
Original dataset from  2001(1) to 2015(2)
Adjusted dataset from  2003(1) to 2014(12)
(...)
```

If the `NA` is not in the extremes but inside the series the routine will warn you but will carry on with some potential adverse effects further down the road. We recommend that you treat the dataset (by interpolating or inputing the missing observations) before running new estimations.

The "fixed term" used in this model is a dummy: `d_lula` is used to control for the first year in power of President Lula in 2003, when interest rates were increased to prevent a significant devaluation of the local currency. This can be checked in the data:



Note that `reer`, the real effective exchange rate, and the other regressors `cpi` and `mpr` look like unit-root processes. A more rigorous approach to testing can be taken by using function `urTable()` from package `macroR` to test for unit roots with the command

```
df <- data.frame( br_month$cpi,br_month$mpr, br_month$reer )
macroR::urTable(df, file="urtests.tex", format="latex")
```

	adf(0)	pp(0)	kpss(0)	adf(1)	pp(1)	kpss(1)
br_month.cpi	0.12	0.47	0.01	0.01	0.01	0.10
br_month.mpr	0.20	0.40	0.01	0.01	0.01	0.10
br_month.reer	0.30	0.41	0.01	0.01	0.01	0.10

Table 2: Unit Root Tests

Results are test p-values for series in levels (0) or in first difference (1).
adf is Augmented Dickey-Fuller Test with H0:series has unit root.
pp is Phillips-Perron Unit Root Test with H0:series has unit root.
kpss is KPSS Test for Level Stationarity with H0:series is stationary.

The ARDL methodology allows the estimation in levels of a common long-term relation between the regressors and the explained variable. In function `coint()` a stationary specification is tested after controlling for the lag of the long-term relation, expressed as `L(coint)`.

To get model details on the coefficients and the usual tests use the traditional `summary()` function

```
summary(m1)
```

```
##
## Time series regression with "zooreg" data:
## Start = 2001(3), End = 2015(1)
##
## Call:
## dynlm::dynlm(formula = formula(fm), data = data, subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93690 -0.11711 -0.01103  0.12217  1.17197
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.402581   0.121226  -3.321  0.00111 **
## L(mpr, 1)    1.700397   0.051446  33.052  < 2e-16 ***
## L(mpr, 2)   -0.733307   0.050578 -14.499  < 2e-16 ***
## cpi          0.069484   0.050493   1.376  0.17072
## L(cpi, 1)   -0.035500   0.050810  -0.699  0.48578
## reer         0.008958   0.005118   1.751  0.08196 .
## L(reer, 1)  -0.002349   0.005380  -0.437  0.66293
## d_lula      -0.541249   0.169134  -3.200  0.00166 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2683 on 159 degrees of freedom
## Multiple R-squared:  0.9969, Adjusted R-squared:  0.9967
## F-statistic:  7209 on 7 and 159 DF,  p-value: < 2.2e-16
```

Function: coint()

To visualize the long-term coefficients use the function `coint()`

```
coint(m1)
```

```
## AutoRegressive Distributed Lag model
## Dependent variable:  mpr
##
## Call:
## mpr ~ +1 + L(mpr, 1) + L(mpr, 2) + cpi + L(cpi, 1) + reer + L(reer,
##     1) + d_lula
## ------------------------------------------------------------
## Short-Run Coefficients. Dependent variable is d(mpr)
## ------------------------------------------------------------
##              Estimate   Std.Err Z value   Pr(>z)
## (Intercept) -0.406627  0.086541  -4.699 2.62e-06 ***
## L(d(mpr))    0.712596  0.048939  14.561  < 2e-16 ***
## d(cpi)       0.035800  0.049685   0.721   0.4712
## d(reer)      0.008433  0.005055   1.668   0.0953 .
## d_lula      -0.059326  0.093817  -0.632   0.5272
## L(coint)    -0.033565  0.006607  -5.080 3.77e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ------------------------------------------------------------
## Long-Run Coefficients (Cointegration Relation)
## ------------------------------------------------------------
##         Estimate   Std.Err Z value   Pr(>z)
## cpi      1.03261   0.22795   4.530  5.9e-06 ***
## reer     0.20082   0.02826   7.106  1.2e-12 ***
## d_lula -16.44606   4.90103  -3.356 0.000792 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the SR coefficients (second panel) come from a model with regressors in first difference, certainly stationary.

The results can also be displayed in LaTeXformat and saved to a file with the options `type` and `file`

```
coint( m1, type="tex", file="m1.tex" )
```

Function: bounds.test()

We recommend that in addition to each individual t-tests for the LR coefficients you test the existence of the cointegration relation with the bounds test. The bounds test checks the existence of a long-term relation with critical values for I(0) and I(1) regressors.

```
bounds.test(m1)
```

```
##
## Bounds Test
## mpr ~ +1 + L(mpr, 1) + L(mpr, 2) + cpi + L(cpi, 1) + reer + L(reer,      1) + d_lula
##
## PSS case 3  ( unrestricted intercert, no trend )
## Regressors (K) 2
##
## d(y_t) = alpha + pi (y_t-1,x_t)' + phi (d(y_t),d(x_t))' + epsilon_t
## Null hypothesis (H0): No long-run relation exist, ie H0:pi=0
##
##          I(0)   I(1)
##   10%   3.17  4.14
##    5%   3.79  4.85
##  2.5%   4.41  5.52
##    1%   5.15  6.36
##
## Wald test to compare the models:
## d(mpr) ~ +1+L(d(mpr)) +d(cpi)+d(reer)+d_lula
## d(mpr) ~ +1+L(d(mpr)) +L(mpr,1)+cpi+reer+d(cpi)+d(reer)+d_lula
##
## F statistic  7.716854
##
## Existence of a Long Term relation is not rejected at 5%
##
##  Long-term coefficients:
##       cpi      reer
## 1.0326097 0.2008177
```

Function: auto.ardl()

The automated model selection process involves choosing the maximum lag for each regressor. If none is informed 1 is assumed.

```
m2 <- auto.ardl( mpr~cpi+prod+reer|d_lula, data=br_month,
                 ymax=2, xmax=c(2,2,2), ic="bic" )
summary(m2)
```

```
##
```

```
## Time series regression with "zooreg" data:
## Start = 2003(3), End = 2014(12)
##
## Call:
## dynlm::dynlm(formula = formula(fm), data = data, subset = subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89576 -0.09533 -0.00244  0.09232  0.73642
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.858712   0.681348   1.260  0.20973
## L(mpr, 1)    1.733216   0.047925  36.165  < 2e-16 ***
## L(mpr, 2)   -0.775263   0.048079 -16.125  < 2e-16 ***
## cpi          0.046507   0.020828   2.233  0.02720 *
## prod        -0.006345   0.003904  -1.625  0.10642
## reer         0.002320   0.001845   1.257  0.21094
## d_lula      -0.561685   0.180928  -3.104  0.00232 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2178 on 135 degrees of freedom
## Multiple R-squared:  0.9976, Adjusted R-squared:  0.9975
## F-statistic:  9532 on 6 and 135 DF,  p-value: < 2.2e-16
```

The selection process involves estimating the best fit for each regressor in the order they are included in the canonical equation. The algorithm will first adjust the best lag for the dependent variable and than proceed to test each regressor following the maximum lags dictated by the `xmax=c(2,0,1)` command that means "test up to the second lag of `cpi`, do not lag `prod` and test only one lag for `reer`". By choosing `verbose=TRUE` you can follow all the tests.

```
m3 <- auto.ardl( mpr~cpi+reer, data=br_month, ymax=2,
                 xmax=c(1,1), verbose=TRUE, case=1 )
```

```
##
## ARDL automatic model selection using bic with ymax= 2 and xmax= 1 1
## Model mpr ~ -1 + L(mpr, 1) + cpi + reer has bic = 276.6923
## Model mpr ~ -1 + L(mpr, 1) + L(mpr, 2) + cpi + reer has bic = 68.17748
## Model mpr ~ -1 + L(mpr, 1) + L(mpr, 2) + cpi + L(cpi, 1) + reer has bic = 71.63689
## Model mpr ~ -1 + L(mpr, 1) + L(mpr, 2) + cpi + reer + L(reer, 1) has bic = 72.84517
## Best model is mpr ~ -1 + L(mpr, 1) + L(mpr, 2) + cpi + reer chosen by bic = 68.17748
```

The selection algorithm relies on the user to choose the case to test. By default the choice is `case=3` (intercept only) but you can specify other cases to test.

Notes on the Algorithm

The function `ardl()` starts by checking the top and bottom of the dataframe for `NA` and exlcude the corresponding rows so that all columns have data. In case there are `NA's` inside the series a warning is emitted. The user should decide on the best way to complete the missing data.

The next step is to build the expanded formula before calling `dynlm()`. The parsed expression is divided in three terms: `lhs` (left hand side) with the dependent variable, `core` with the variable term(s) and `suffix` with the fixed term(s). To map the coefficients of the canonical form into the expanded form we use the `coeff_map` vector: the content of this vector is `"0"` for the lhs (y) and `"1"` for the first element of `rhs` and so on until the `K+KX` term is reached, `K` is the number of variable terms and `KX` is the number of fixed terms.

For example: the canononical form `y~x1+x2|x3` with `case=3`, `ylag=2` and `xlag=(3,1)` generates the extended form `y~+1+L(y,1)+L(y,2)+x1+L(x1,1)+L(x1,2)+L(x1,3)+x2+L(x2,1)+x3` with mapping `coeff_map == "-1" "0" "0" "1" "1" "1" "1" "2" "2" "2" "3"`. Note that Intercept and Trend are marked with "-1" as a placeholder only.

Once this estimation is done, the `dynlm` object is extended with fields for the long-run (LR) and short-run (SR) coefficients, the case number and the cointegration relation. The LR coefficients are calculated as indicated by M. H. Pesaran and Shin (1999) and a cointegrating relation is built to reestimate the model, now controlling for the long-term, so that the new coefficients are the SR coefficients.

Future Developments

1. Support for restricted coefficient estimation (ML) and cases 2 and 4.
2. Support `plot()` showing actual and fitted data, residual and the cointegration relation.
3. Function `coint()` should present test results for residual autoregression and heterocedasticity (and R2, F, etc…).
4. Function `auto.ardl()` should adjust sample size to the same for all model comparisons.
5. Support to 2SLS estimation with instruments, for ex. `ardl( y ~ x1 + x2, instrument=list(x1,x3,x4) )` where `x3` and `x4` are instruments for `x2`.
6. Support to structural models with time varying parameters (TVP) implemented by Kalman Filter.

Bibliography

Hassler, Uwe, and Jürgen Wolters. 2005. *Autoregressive distributed lag models and cointegration.* Discussion Papers 2005/22. https://ideas.repec.org/p/zbw/fubsbe/200522.html.

Pesaran, M, Yongcheol Shin, and Richard Smith. 2001. "Bounds Testing Approaches to the Analysis of Level Relationships." *Journal of Applied Econometrics* 16 (3): 289–326. http://EconPapers.repec.org/RePEc:jae:japmet:v:16:y:2001:i:3:p:289-326.

Pesaran, M. Hashem, and Yongcheol Shin. 1999. “An Autoregressive Distributed-Lag Modelling Approach to Cointegration Analysis.” In *Econometrics and Economic Theory in the 20th Century*, edited by Steinar Strøm, 371–413. Cambridge University Press. http://dx.doi.org/10.1017/CCOL521633230.011.