Package 'oosanalysis'

February 1, 2016

Type Package

Title Pseudo out-of-sample forecasting

Date 2011-12-29

Author Gray Calhoun

Maintainer Gray Calhoun <gray@clhn.org>

Description Functions for out-of-sample comparisons of forecasting models

Depends R(>= 2.10), MASS

Suggests testthat

LazyLoad yes

LazyData yes

Version 0.4.0

R topics documented:

`bootindex_circularblock`

Indices to induce block bootstraps

Description

These functions generate the random indices necessary to implement the moving blocks bootstrap (Kunsch, 1989) and the circular blocks bootstrap (Politis and Romano, 1994)

Usage

```
bootindex_movingblock(nobs, blocklength)
bootindex_circularblock(nobs, blocklength)
bootindex_stationary(nobs, blocklength)
```

Arguments

`nobs`	The length of each bootstrap process.
`blocklength`	The block length or, in the case of the stationary bootstrap, the expected block length.

Details

Both the moving blocks bootstrap and the circular blocks bootstrap resample from the sequence X = X[1],...,X[n] by drawing length `blocklength` consecutive observations repeatedly and then pasting the blocks until the resampled sequence has the same length as the original. The blocks are drawn independently of each other. The circular block bootstrap allows (for example) X[n-1], X[n], X[1], X[2] to be a valid block of length 4, and the moving blocks bootstrap does not. The functions `bootindex_movingblock` and `bootindex_circularblock` give the indices that induce the bootstrap, so for example we get a particular circular block bootstrap draw of `X` with block length 8 from the command `X[bootindex_movingblock(length(X), 8)]`.

Value

A vector of indices that corresponds to a single bootstrap draw.

Author(s)

Gray Calhoun `<gcalhoun@iastate.edu>`

References

Calhoun, G. 2011, Documentation appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Kunsch, H. R. 1989, The Jackknife and the Bootstrap for general stationary observations. *Annals of Statistics*, **17**(3), pages 1217–1241.

Liu, R. Y. and Kesar, S. 1992, Moving blocks Jackknife and Bootstrap capture weak dependence, in R. LePage and L. Billard, editors, *Exploring the limits of Bootstrap*, John Wiley, pages 225–248.

Politis, D. N. and Romano, J. P. 1992, A circular block-resampling procedure for stationary data, in R. LePage and L. Billard, editors, *Exploring the limits of Bootstrap*, John Wiley, pages 263–270.

See Also

boot

Examples

```
## Example of hypothesis test that mean = 0
nobs <- 200
nboot <- 299
level <- .1
X <- 2 + arima.sim(n = nobs, list(ma = c(0.5)))

naive <- replicate(nboot, mean(X[sample(1:nobs, nobs, replace = TRUE)])) - mean(X)
smart1 <- replicate(nboot, mean(X[bootindex_circularblock(nobs, 5)])) - mean(X)
smart2 <- replicate(nboot, mean(X[bootindex_movingblock(nobs, 5)])) - mean(X)
smart3 <- replicate(nboot, mean(X[bootindex_stationary(nobs, 5)])) - mean(X)

## corresponding critical values
quantile(naive, 1 - level)
quantile(smart1, 1 - level)
quantile(smart2, 1 - level)
quantile(smart3, 1 - level)

## Not run:
mc <- replicate(300, {
  X <- arima.sim(n = nobs, list(ma = c(0.5)))
  naive <- replicate(nboot, mean(X[sample(1:nobs, nobs, replace = TRUE)])) - mean(X)
  smart <- replicate(nboot, mean(X[bootindex_circularblock(nobs, 5)])) - mean(X)
  c(naive = mean(X) >= quantile(naive, 1 - level),
    smart = mean(X) >= quantile(smart, 1 - level))
  })
rowMeans(mc)

## End(Not run)
```

---

clarkwest *Clark and West's (2006, 2007) Out-of-Sample Test*

Description

Functions to calculate Clark and West's (2006, 2007) approximately normal OOS statistic.

Usage

```
clarkwest(null, alt, dataset, R, vcv = var,
          window = c("rolling", "recursive", "fixed"))

clarkwest_calculation(target, null.forecast, alt.forecast, vcv)
```

Arguments

null	A function that takes a subset of the data dataset as its argument and returns an object with a predict method. This function generates the benchmark forecast.

`alt`	A second function that takes a subset of the data `dataset` as its argument and returns an object with a `predict` method. This function generates the alternative forecast.
`dataset`	A data frame.
`R`	An integer, the size of the training sample.
`vcv`	A function to calculate the asymptotic variance of the OOS average.
`window`	A character that indicates the window strategy for OOS estimation.
`target`	A vector containing the values of the predictand.
`null.forecast`	A vector containing the values of the benchmark forecast.
`alt.forecast`	A vector containing the values of the alternative forecast.

Details

Both of these functions implement Clark and West's (2006, 2007) "corrected" out-of-sample tests. The idea behind their tests is that using a fixed-length rolling window, as in Giacomini and White (2006), ensures that the OOS average is asymptotically normal. In Giacomini and White, though, the OOS average is not centered at the expected difference in the MSE of the pseudo-true forecasting models, so Clark and West introduce an adjustment so that their statistic is centered correctly. Be aware that Clark and West's adjustment is provably correct for the fixed or rolling windows when `R` is small *and the benchmark model is not estimated*, though Clark and West's (2007) simulations indicate that it performs well for estimated benchmarks for some DGPs. See Calhoun (2011) for an asymptotically normal OOS statistic that allows the benchmark to be estimated. The function allows users to choose the "recursive" estimation strategy because it is popular in practice, but be careful.

`clarkwest_calculation` does all of the algebra and `clarkwest` is a convenient interface to it that calculates the forecasts automatically.

Value

Both functions return the same thing, a list with elements

`mu`	an estimate of the corrected OOS mean,
`avar`	the asymptotic variance of the corrected OOS average,
`pvalue`	the p-value associate with the one-sided OOS test.

Author(s)

Gray Calhoun `<galhoun@iastate.edu>`

References

Calhoun, G. 2011, An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Calhoun, G. 2011, Documentation appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Clark, T. E., West, K. D. 2006, Using out-of-sample mean squared prediction errors to test the martingale difference hypothesis. *Journal of Econometrics*, **135**(1): 155–186.

Clark, T. E., West, K. D. 2007, Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, **138**(1): 291–311.

See Also

dmw_calculation, mixedwindow, mccracken_criticalvalue, recursive_forecasts, predict

Examples

```
x <- rnorm(100)
d <- data.frame(y = x + rnorm(100), x = x)
R <- 70

model1 <- function(d) lm(y ~ 1, data = d)
model2 <- function(d) lm(y ~ x, data = d)

clarkwest(model1, model2, d, R, window = "rolling")
```

Classes *Some classes to simplify making predictions*

Description

This file defines some classes and methods that are used to make some of the predictions in the corresponding paper's empirical exercise (Calhoun, 2011).

Usage

```
CT(model)
Aggregate(model.list, fn)
## S3 method for class 'CT'
predict(object, newdata,...)
## S3 method for class 'Aggregate'
predict(object, newdata,...)

HasMethod(object, method.name)
```

Arguments

model	An object with a "predict" method.
model.list	A list of objects with "predict" methods.
fn	A function that can take and aggregate a vector: "mean," for example.
object	A variable belonging to an S3 class.
newdata	A new data set to use to create the new forecasts.
...	For "predict.CT", additional arguments to pass to the underlying "predict" method. For "predict.Aggregate", additional arguments to pass to the individual "predict" methods.
method.name	A character vector giving the names of different S3 methods.

Value

"CT" and "Aggregate" return objects with (S3) classes "CT" and "Aggregate" respectively. The "predict" methods each return a single forecast. "HasMethod" returns a logical vector with the same length as "method.name" indicating whether "object" has each method defined.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

Examples

```
olddata <- data.frame(y = rnorm(30), x = rnorm(30))
newdata <- data.frame(y = rnorm(3), x = rnorm(3))

m1 <- lm(y ~ 1, data = olddata)
m2 <- lm(y ~ x, data = olddata)

m3 <- CT(m2)
m4 <- Aggregate(list(m1, m2, m3), median)

predict(m3, newdata)
predict(m4, newdata)

HasMethod(m1, c("plot", "print", "predict", "median"))
```

---

dmw_mse *Diebold-Mariano-West out-of-sample t-test*

Description

The Diebold-Mariano-West OOS t-test can be used to compare population forecasting models under some fairly restrictive circumstances (see West, 2006). The forecast are assumed to be constructed using a fixed, recursive, or rolling estimation window and depend on the estimated coefficients $\hat{\beta}_t$. The function dmw_calculation takes as arguments the matrices and vectors that West (1996) and West and McCracken (1998) use to represent the asymptotic distribution of this statistic and just assembles the mean and variance components of the statistic. dmw_mse is a basic convenience wrapper for the common use case: squared error loss with least squares forecasts. The mixedwindow functions implement the asymptotically normal OOS test statistics proposed by Calhoun (2011).

Usage

```
dmw_mse(null, alt, dataset, R, vcv = var,
        window = c("recursive", "rolling", "fixed"))

dmw_calculation(f, h, R, vcv, tBtF = NULL, pi = noos / R,
                window = c("recursive", "rolling", "fixed"))

mixedwindow(null, alt, dataset, R, vcv = var,
            window = c("rolling", "fixed"), pimethod = "estimate")

mixedbootstrap(null, alt.list, dataset, R, nboot, blocklength,
               vcv = var, window = c("rolling", "fixed"),
               bootstrap = c("moving", "circular", "stationary"),
               pimethod = "estimate")
```

Arguments

`null` A function that takes a subset of the data `dataset` as its argument and returns an object with a `predict` method. This function generates the benchmark forecast.

`alt` A second function that takes a subset of the data `dataset` as its argument and returns an object with a `predict` method. This function generates the alternative forecast.

`alt.list` A list of functions that would be valid as `alt`

`dataset` A data frame.

`R` An integer, the size of the training sample. The asymptotic theory assumes that `R` is small.

`f` A vector containing the OOS observations

`h` A matrix containing something like (for OLS using the obvious notation) $x_t \varepsilon_t$ for t ranging over the OOS period.

`tBtF` A vector that represents $B'F'$ in West's (1996) notation. This term captures the uncertainty introduced by estimating the unknown model coefficients; if the coefficients are known or imposed, instead of estimated, set this argument to `NULL`

`pi` A numeric scalar, the ratio of the number of out-of-sample observations to the number of training sample observations. `noos` is defined in the body of the function as `length(f)`.

`window` A character string indicating which window strategy was used to generate the OOS observations. For the `mixedwindow` functions, this is the window strategy for OOS estimation for the alternative model[s] since the benchmark model is always estimated with the recursive scheme.

`nboot` An integer, the number of bootstrap replications.

`blocklength` An integer, the length of the blocks for the moving or circular block bootstraps.

`vcv` A function to calculate the asymptotic variance of the OOS average.

`pimethod` Indicates whether Pi (= lim P/R) should be estimated as P/R (`pimethod = "estimate"`) or set to the theoretical limit of infinity (`pimethod = "theory"`).

`bootstrap` Indicates whether to do the moving blocks bootstrap (MBB) (Kunsch, 1989 and Liu and Singh, 1992), circular blocks bootstrap (CBB) (Politis and Romano, 1992), or stationary bootstrap (Politis and Romano, 1994)

Details

Calhoun's (2011) mixed window OOS test is a modification of Clark and West's (2006, 2007) that uses a recursive window for the benchmark model to ensure that the OOS average is mean zero and asymptotically normal. `mixedwindow` compares a pair of models and `mixedbootstrap` implements the bootstrap used for multiple comparisons.

Value

`dmw_mse` and `dmw_calculation` each return a list containing the following elements:

`mu` The OOS average,

`avar` The asymptotic variance of the OOS average.

`mixedwindow` returns a list with the following elements:

`mu`	The estimated OOS average, which includes the adjustment for correct asymptotic centering
`avar`	An estimate of the asymptotic variance of the OOS average
`pvalue`	The p-value of the test that the two models have equal population MSE against the one-sided alternative that the alternative model is more accurate.

`mixedbootstrap` returns an `length(alt.list)` by `nboot` matrix that contains the resampled values of the OOS t-test based on `mixedwindow`. These are the values of the t-statistic and not the test's p-values.

Author(s)

Gray Calhoun `<gcalhoun@iastate.edu>`

References

Calhoun, G. 2011, An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Calhoun, G. 2011, Supplemental appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Clark, T. E., West, K. D. 2006, Using out-of-sample mean squared prediction errors to test the martingale difference hypothesis. *Journal of Econometrics*, **135**(1): 155–186.

Clark, T. E., West, K. D. 2007, Approximately normal tests for equal predictive accuracy in nested models. *Journal of Econometrics*, **138**(1): 291–311.

Diebold, F. X. and Mariano, R. S. 1995, Comparing predictive accuracy. *Journal of Business and Economic Statistics*, **138**(1): 253–263.

Kunsch, H. R. 1989, The Jackknife and the Bootstrap for general stationary observations. *Annals of Statistics*, **17**(3), pages 1217–1241.

Liu, R. Y. and Kesar, S. 1992, Moving blocks Jackknife and Bootstrap capture weak dependence, in R. LePage and L. Billard, editors, *Exploring the limits of Bootstrap*, John Wiley, pages 225–248.

Politis, D. N. and Romano, J. P. 1992, A circular block-resampling procedure for stationary data, in R. LePage and L. Billard, editors, *Exploring the limits of Bootstrap*, John Wiley, pages 263–270.

Politis, D. N. and Romano, J. P. 1994, The Stationary Bootstrap. *Journal of the American Statistical Association*, **89**(428), pages 1303-1313.

West, K. D. 1996, Asymptotic inference about predictive ability. *Econometrica*, **64**(5): 1067–1084.

West, K. D. 2006, Forecast evaluation, in G. Elliott, C. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, volume 1, pages 99–134. Elsevier.

West, K. D. and McCracken, M. W. 1998, Regression-based tests of predicitve ability. *International Economic Review*, **39**(4):817–840.

See Also

`clarkwest`, `mccracken_criticalvalue`, `recursive_forecasts`, `predict`, `boot`

Examples

```
x <- rnorm(100)
d <- data.frame(y = x + rnorm(100), x = x)
R <- 70
oos <- 71:100

error.model1 <- d$y[oos] - predict(lm(y ~ 1, data = d[-oos,]),
                                   newdata = d[oos,])
error.model2 <- d$y[oos] - predict(lm(y ~ x, data = d[-oos,]),
                                   newdata = d[oos,])
# test that the two models have equal population MSE.  Note that F = 0
# in this setting.
estimates <-
  dmw_calculation(error.model1^2 - error.model2^2,
                  cbind(error.model1, error.model2, error.model2 * x),
                  R = R, vcv = var)
# calculate p-value for a one-sided test
pnorm(estimates$mu * sqrt(length(oos) / estimates$avar))


n <- 30
R <- 5
d <- data.frame(y = rnorm(n), x1 = rnorm(n), x2 = rnorm(n))
model0 <- function(d) lm(y ~ 1, data = d)
model1 <- function(d) lm(y ~ x1, data = d)
model2 <- function(d) lm(y ~ x2, data = d)
model3 <- function(d) lm(y ~ x1 + x2, data = d)

mixedwindow(model0, model1, d, R, var, window = "rolling")

mixedbootstrap(model0, list(m1 = model1, m2 = model2, m3 = model3),
               d, R, 199, 7, var, "fixed", "circular")
```

---

extract_target — *Convenience function to extract data from a model*

Description

Convenience function to extract data from a model.

Usage

```
extract_target(model, dataset)
extract_predictors(model, dataset)
```

Arguments

`model`	A function that takes a data frame as its argument and returns an object that has a `terms` method.
`dataset`	A data frame.

Details

text

Value

For `extract_target`, a vector that contains the values from `dataset` of the dependent variable specified in `model`. For `extract_predictors`, the same thing but now the `model.matrix` of the predictors.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

References

Calhoun, G. 2011, Documentation appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

See Also

`mixedwindow`, `clarkwest`, `terms`, `model.matrix`

Examples

```
model <- function(d) lm(y ~ x, data = d)
dataset <- data.frame(y = rnorm(10), x = rnorm(10))


all.equal(extract_target(model, dataset), dataset$y,
          check.attributes = FALSE)



all.equal(extract_predictors(model, dataset),
          cbind(1, dataset$x), check.attributes = FALSE)
```

McCrackenData *Critical values for oos-t test from McCracken (2007)*

Description

This data set reproduces tables 1, 2, and 3 from McCracken (2007). The function `mccracken_criticalvalue` can be used to retrieve specific values.

Usage

```
McCrackenData
```

Format

A list of arrays.

References

McCracken, M. 2007, Asymptotics for out of sample tests of Granger causality. *Journal of Econometrics*, **140**(2): 719-752.

See Also

mccracken_criticalvalue

---

mccracken_criticalvalue *Returns McCracken's (2007) oos-t critical values*

Description

This function retrieves the critical values for the oos-t test statistic derived by McCracken (2007) for nested models.

Usage

```
mccracken_criticalvalue(pi, k2, confidence, window = c("recursive", "rolling", "fixed"))
```

Arguments

pi	P/R where P is the number of out-of-sample observations and R is the size of the estimation window. Note that this is rounded to the closest value in McCracken's table; if you want to interpolate between listed values, you must do so manually.
k2	The number of additional regressors in the larger model.
confidence	One minus the asymptotic nominal size of the test; this must be 0.90, 0.95, or 0.99.
window	The OOS window scheme.

Value

Returns a single numeric value, the appropriate critical value for the test.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

References

McCracken, M. 2007, Asymptotics for out of sample tests of Granger causality. *Journal of Econometrics*, **140**(2): 719-752.

See Also

clarkwest, recursive_forecasts, dmw_mse, dmw_calculation

Examples

```
mccracken_criticalvalue(.4, 5, .9, "rolling")
mccracken_criticalvalue(.4, 5, .9, "recursive")
mccracken_criticalvalue(.4, 5, .9, "fixed")
```

recursive_forecasts *Pseudo out-of-sample forecasts*

Description

Creates a sequence of pseudo out-of-sample forecasts.

Usage

```
recursive_forecasts(model, dataset, R,
                    window = c("recursive", "rolling", "fixed"),...)
```

Arguments

`model`	A function that takes `dataset` as its first argument. `model` must return an object with a `predict` method.
`dataset`	A data frame with more than `R` observations
`R`	An integer: the size of the estimation window.
`window`	One of "rolling", "recursive", or "fixed" describing the estimation strategy
`...`	Additional arguments to pass to `model`

Details

Uses `model` to create a sequence of forecasts or forecast errors for observations `R+1`,...,`nrow(dataset)`. For the "rolling" window, each forecast comes from the model estimated with the previous `R` observations. For the "recursive" window, each forecast uses all of the previous observations. And for the "fixed" window, each forecast uses the first `R` observations.

Value

A vector of length `nrow(dataset) - R`, containing the forecasts.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

References

Calhoun, G. 2011, Documentation appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Examples

```
d <- data.frame(x = rnorm(15), y = rnorm(15))
ols <- function(d) lm(y ~ x, data = d)
## Basic Usage:
recursive_forecasts(ols, d, 4, "recursive")

## Illustrate different estimation windows by comparing forecasts for
## observation 11 (note that the forecast for observation 11 will be the
## 7th element that apply.oos returns in this example)
```

```
newd <- d[11,]


all.equal(predict(lm(y ~ x, data = d[7:10,]), d[11,]),
          recursive_forecasts(ols, d, 4, "rolling")[7])



all.equal(predict(lm(y ~ x, data = d[1:10,]), d[11,]),
          recursive_forecasts(ols, d, 4, "recursive")[7])



all.equal(predict(lm(y ~ x, data = d[1:4,]), d[11,]),
          recursive_forecasts(ols, d, 4, "fixed")[7])
```

---

rvar *Generate pseudo-random data from a Vector Autoregression*

Description

This function generates data from a user-specified Vector Autoregression (VAR)

Usage

```
rvar(nobs, coefficients, intercept, vcv, nburn = 1000,
     y0 = matrix(0, nlag, neq))
```

Arguments

nobs	Integer; the number of observations to generate.
coefficients	A list of numeric vectors; each vector specifies the coefficients of one of the VAR equations.
intercept	A numeric vector containing the intercepts.
vcv	A numeric matrix representing the variance-covariance matrix of the innovations.
nburn	An integer, the number of draws to discard before generating draws from the VAR.
y0	A matrix containing the inital values of the series.

Details

Let β_1 represent the first element of the coefficients list and suppose that there are k equations with l lags. The first equation of the DGP is

$$y_{1t} = \mu_1 + (y_{1,t-1}, \dots, y_{1,t-l}, y_{2,t-1}, \dots, y_{2,t-l}, \dots y_{k,t-1}, \dots, y_{k,t-l})'\beta_1 + \varepsilon_{1t}$$

.

Value

A data frame with nobs rows containing the generated series and all of their lags. This is redundant information, but makes it easier to use these results directly in a regression model.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

Examples

```
d <- rvar(10000, list(a = c(0.5, 0, 0.2, 0.1),
                      b = c(0.1, 0.2, 0.5, 0)),
          c(4, 6), diag(2))

lm(a ~ aL1 + aL2 + bL1 + bL2, data = d)
lm(b ~ aL1 + aL2 + bL1 + bL2, data = d)
```

`stepm` *Romano and Wolf's (2005) StepM*

Description

Implements Romano and Wolf's StepM, a stepdown procedure for multiple hypothesis testing.

Usage

```
stepm(teststatistics, bootmatrix, lefttail, righttail)
```

Arguments

`teststatistics`	A numeric vector, containing test statistics.
`bootmatrix`	A matrix with `length(teststatistics)` rows and an arbitrary number of columns. Each column is a different draw from the bootstrap-induced distribution of `teststatistics`.
`lefttail`	The mass to be left in the left tail of the distribution. Setting this to `NA` imposes a one-sided alternative.
`righttail`	The mass to be left in the right tail of the distribution. Setting this to `NA` imposes a one-sided alternative.

Details

This function assumes that each element of `teststatistics` tests (say) $\mu_i = 0$ against the alternative $mu_i > 0$, for i = 1,...,`length(teststatistics)`. Romano and Wolf's (2005) StepM procedure estimates a critical value such that the probability that it is smaller than at least one statistic corresponding to a true null hypothesis is controlled at level `level`.

Value

A list with the following elements:

`criticalvalues`	The estimated critical values.
`rejected`	A logical vector indicating which statistcs fall outside the critical values.

Author(s)

Gray Calhoun <gcalhoun@iastate.edu>

References

Calhoun, G. 2011, Documentation appendix: An asymptotically normal out-of-sample test of equal predictive accuracy for nested models. Unpublished manuscript.

Calhoun, G. 2012, A comment on "Stepwise multiple testing as formalized data snooping." Unpublished manuscript.

Romano, J. P., and Wolf, M. 2005, Stepwise multiple testing as formalized data snooping. *Econometrica*, **73**(4), pages 1237–1292.

See Also

boot

Examples

```
n <- 50
nboot <- 99
d <- data.frame(x1 = rnorm(n), x2 = rnorm(n) + 1, x3 = rnorm(n))

dottests <- function(dataset)
  sapply(dataset, function(x) t.test(x)$statistic)

stepm(teststatistics = dottests(d),
      bootmatrix = replicate(nboot, dottests(d[sample(1:n, n, replace = TRUE),])),
      lefttail = NA, righttail = 0.05)
```

Index