Package ‘multDM’

May 18, 2018

Type Package

Title Multivariate Version of the Diebold-Mariano Test

Version 1.0

Date 2018-05-18

Author Krzysztof Drachal [aut, cre] (Faculty of Economic Sciences, University of Warsaw, Poland)

Maintainer Krzysztof Drachal <kdrachal@wne.uw.edu.pl>


License GPL-3

LazyData TRUE

URL https://CRAN.R-project.org/package=multDM

Note Research funded by the Polish National Science Centre grant under the contract number DEC-2015/19/N/HS4/00205.

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-18 14:04:45 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM.test</td>
<td>2</td>
</tr>
<tr>
<td>d_t</td>
<td>3</td>
</tr>
<tr>
<td>loss</td>
<td>4</td>
</tr>
<tr>
<td>MDM.selection</td>
<td>5</td>
</tr>
<tr>
<td>MDM.test</td>
<td>6</td>
</tr>
<tr>
<td>MDMforecasts</td>
<td>7</td>
</tr>
<tr>
<td>oilforecasts</td>
<td>8</td>
</tr>
<tr>
<td>print.MDM</td>
<td>9</td>
</tr>
<tr>
<td>TB_MA</td>
<td>10</td>
</tr>
</tbody>
</table>

Index 12
Computes Diebold-Mariano Test for the Equal Predictive Accuracy.

**Description**

This function computes Diebold-Mariano test for the equal predictive accuracy. The null hypothesis of this test is that two forecasts have the same accuracy. The alternative hypothesis can be specified as "Both forecasts have different accuracy", "The first forecast is less accurate than the second forecast", or "The first forecast is more accurate than the second forecast".

**Usage**

```r
dm.test(f1, f2, y, loss.type = "SE", h, c = FALSE, H1 = "same")
```

**Arguments**

- `f1`: vector of the first forecast
- `f2`: vector of the second forecast
- `y`: vector of the real values of the modelled time-series
- `loss.type`: method to compute the loss function, `loss.type = "SE"` will use squared errors, `loss.type = "AE"` will use absolute errors, `loss.type = "SPE"` will use squared proportional error (useful if errors are heteroskedastic), if `loss.type` will be specified as some numeric, then the function of type `exp(loss.type*errors)-1-loss.type*errors` will be used (useful when it is more costly to underpredict than to overpredict), if not specified `loss.type = "SE"` is used
- `h`: numeric denoting that the forecast h-steps ahead are evaluated, if not specified `h=1` is used
- `c`: logical indicating if Harvey-Leybourne-Newbold correction for small samples should be used, if not specified `c=FALSE` is used
- `H1`: alternative hypothesis, `H1 = "same"` for "both forecasts have different accuracy", `H1 = "more"` for "the first forecast is more accurate than the second forecast", `H1 = "less"` for "the first forecast is less accurate than the second forecast", if not specified `H1 = "same"` is taken

**Value**

- `class` htest object, `list` of
  - `statistic`: test statistic
  - `parameter`: h, forecast horizon used
  - `alternative`: alternative hypothesis of the test
  - `p.value`: p-value
  - `method`: name of the test
  - `data.name`: names of the tested time-series
References


Examples

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
DM.test(f1=forecasts[,1], f2=forecasts[,2], y=ts, loss="SE", h=1, c=FALSE, H1="same")
```

<table>
<thead>
<tr>
<th>d_t</th>
<th>Computes Loss Differential.</th>
</tr>
</thead>
</table>

Description

This function computes loss differential, i.e., differences between losses from \( k + 1 \)-th and \( k \)-th models.

Usage

```
d_t(e)
```

Arguments

- `e` matrix of loss functions, columns correspond to time index, and rows to different models

Value

matrix of loss differentials

References


Examples

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
l <- loss(realized=ts, evaluated=forecasts, loss.type="SE")
d <- d_t(l)
```
Computes Loss Function.

Description

This function computes various loss functions for given realized values of time-series and a collection of forecasts.

Usage

```
loss(realized, evaluated, loss.type)
```

Arguments

- `realized`: vector of the real values of the modelled time-series
- `evaluated`: matrix of the forecasts, columns correspond to time index, rows correspond to different models
- `loss.type`: method to compute the loss function, `loss.type="SE"` will use squared errors, `loss.type="AE"` will use absolute errors, `loss.type="SPE"` will use squared proportional error (useful if errors are heteroskedastic), if `loss.type` will be specified as some numeric, then the function of type `exp(loss.type*errors)-1-loss.type*errors` will be used (useful when it is more costly to underpredict realized than to overpredict)

Value

`matrix` with columns corresponding to time index and rows to different models

References


Examples

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
l <- loss(realized=ts, evaluated=forecasts, loss.type="SE")
```
**MDM.selection**

Selects Models with Outstanding Predictive Ability basing on Multivariate Diebold-Mariano Test.

**Description**

This function selects models with outstanding predictive ability basing on multivariate Diebold-Mariano test `MDM.test`.

**Usage**

```r
MDM.selection(realized, evaluated, q, alpha, statistic="Sc", loss.type="SE")
```

**Arguments**

- `realized` vector of the real values of the modelled time-series
- `evaluated` matrix of the forecasts, columns correspond to time index, rows correspond to different models
- `q` numeric indicating a lag length beyond which we are willing to assume that the autocorrelation of loss differentials is essentially zero
- `alpha` numeric indicating a significance level for multivariate Diebold-Mariano tests
- `statistic` statistic for the basic version of the test, and statistic="Sc" for the finite-sample correction, if not specified statistic="Sc" is used
- `loss.type` method to compute the loss function, loss.type="SE" will use squared errors, loss.type="AE" will use absolute errors, loss.type="SPE" will use squared proportional error (useful if errors are heteroskedastic), if loss.type will be specified as some numeric, then the function of type `exp(loss.type*errors)-1-loss.type*errors` will be used (useful when it is more costly to underpredict realized than to overpredict), if not specified loss.type="SE" is used

**Value**

class MDM object, list of

- `outcomes` matrix with mean losses for the selected models, statistics corresponding to losses differentials and ranking of these statistics
- `p.value` numeric of p-value from the procedure, i.e., p-value of multivariate Diebold-Mariano test from the last step
- `alpha` alpha, i.e., the chosen significance level
- `eliminated` numeric indicating the number of eliminated models

**References**

Examples

```r
data(MDMforecasts)
ns <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
MDM.selection(realized=ts,evaluated=forecasts,q=10,alpha=0.1,statistic="S",loss.type="AE")
```

**MDM.test**

*Computes Multivariate Diebold-Mariano Test for the Equal Predictive Accuracy of Two or More Non-nested Forecasting Models.*

Description

This function computes multivariate Diebold-Mariano test for the equal predictive accuracy of two or more non-nested forecasting models. The null hypothesis of this test is that the evaluated forecasts have the same accuracy. The alternative hypothesis is that Equal predictive accuracy (EPA) does not hold.

Usage

```r
MDM.test(realized,evaluated,q,statistic="Sc",loss.type="SE")
```

Arguments

- **realized** vector of the real values of the modelled time-series
- **evaluated** matrix of the forecasts, columns correspond to time index, rows correspond to different models
- **q** numeric indicating a lag length beyond which we are willing to assume that the autocorrelation of loss differentials is essentially zero
- **statistic** statistic for the basic version of the test, and statistic="Sc" for the finite-sample correction, if not specified statistic="Sc" is used
- **loss.type** method to compute the loss function, loss.type="SE" will use squared errors, loss.type="AE" will use absolute errors, loss.type="SPE" will use squared proportional error (useful if errors are heteroskedastic), if loss.type will be specified as some numeric, then the function of type exp(loss.type*errors)-1-loss.type*errors will be used (useful when it is more costly to underpredict realized than to overpredict), if not specified loss.type="SE" is used

Value

- class htest object, list of
  - **statistic** test statistic
  - **parameter** q, a lag length
  - **alternative** alternative hypothesis of the test
  - **p.value** p-value
  - **method** name of the test
  - **data.name** names of the tested objects
**References**


**Examples**

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
MDM.test(realized=ts, evaluated=forecasts, q=10, statistic="S", loss.type="AE")
```

---

**MDMforecasts**  
*SAMPLE DATA.*

---

**Description**

Sample artificial data.

**Usage**

```r
data(MDMforecasts)
```

**Format**

MDMforecasts is **list** object such that

- **MDMforecasts$ts** is **vector** of time-series which is of interest to model
- **MDMforecasts$forecasts** is **matrix** of 20 different forecasts of ts from 20 different forecasting models, each row represents different forecast and time is indexed by columns

**Examples**

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
MDM.test(realized=ts, evaluated=forecasts, q=10, statistic="S", loss.type="AE")
```
Sample Data from Crude Oil Price Forecasting.

Description
Forecasts obtained from various methods applied to crude oil price.

Usage
data(oilforecasts)

Format
oilforecasts is matrix object such that its rows correspond to forecasts from various methods, i.e.,

- REALIZED is the forecasted time-series,
- DMA.DOW is the forecast from Dynamic Model Averaging with the dynamic Occam’s window,
- BMA.DOW is the forecast from Bayesian Model Averaging with the dynamic Occam’s window,
- DMA.1V is the forecast from Dynamic Model Averaging applied only to one-variable models,
- BMA.1V is the forecast from Bayesian Model Averaging applied only to one-variable models,
- DMS.1V is the forecast from Dynamic Model Selection applied only to one-variable models,
- BMS.1V is the forecast from Bayesian Model Selection applied only to one-variable models,
- TVP is the forecast from Time-Varying Parameters regression,
- LASSO is the forecast from LASSO regression,
- RIDGE is the forecast from RIDGE regression,
- DYN.EL.NET is the forecast from the elastic net regression, with the elastic net mixing parameter changing with time index,
- LARS is the forecast from the least-angle regression,
- B.LASSO is the forecast from the Bayesian LASSO regression,
- B.RIDGE is the forecast from the Bayesian RIDGE regression,
- ARIMA is the forecast from the best ARIMA model according to AIC,
- NAIVE is the naive forecast, i.e., the last observation is taken as a one-step ahead prediction,
- MA is the moving average.

Details
The data were taken from Juvenal and Petrella (2015). They cover the period between 1971 and 2009, and are in quarterly frequency. Time-series with missing observations were excluded from the original data set, resulting finally in 127 explanatory variables, instead of 150 in the original data set. In particular, the excluded time-series are the ones which start date is after 1971. The dependent time-series is the average world price of oil taken in logarithmic differences. The independent time-series represent various stationarity transformations of macroeconomic and financial variables of the
G7 countries and from the oil market, global economic activity and various commodity prices. The
details of the original data set are given in the paper by Juvenal and Petrella (2015). The forecasting
with various models, based on this data set, was done by the author of this package, just to provide
some more concrete example set of forecasts. The independent variables were taken in the first
lags. The forgetting parameters in DMA/DMS models were set to 0.99, resulting in the effective
rolling window size of 100. Therefore, such a window was taken for the moving average. LASSO
and RIDGE (also in the Bayesian versions), the elastic net, the least-angle regression and ARIMA
models were estimated in rolling windows of the size of 100 observations. First 100 observations
were excluded, and oilforecasts consists of the remaining last observations. The estimations
were done with the following packages fDMA, forecast, glmnet, lars and monomvn.

References

Drachal, K. 2017. fDMA: Dynamic Model Averaging and Dynamic Model Selection for continuous
outcomes. https://CRAN.R-project.org/package=fDMA

Friedman, J., Hastie, T., Tibshirani, R. 2010. Regularization paths for generalized linear models via

Gramacy, R.B. 2017. monomvn: Estimation for Multivariate Normal and Student-t Data with
Monotone Missingness. https://CRAN.R-project.org/package=monomvn

CRAN.R-project.org/package=lars

Hyndman, R.J., Khandakar. Y. 2008. Automatic time series forecasting: the forecast package for

612–649.

Examples

```r
data(oilforecasts)
ts <- oilforecasts[1,]
forecasts <- oilforecasts[-1,]
mdm.selection(realized=ts,evaluated=forecasts,q=8,alpha=0.1,statistic="Sc",loss.type="SE")

print.MDM
```

```
print.MDM
Prints MDM Object.
```

Description

The function prints selected outcomes obtained from *MDM.selection*.

Usage

```r
## S3 method for class 'MDM'
print(x, ...)
```
Arguments

- \( x \) an object of \( \text{MDM} \) class
- \( \ldots \) not used

Details

The function prints models with outstanding predictive ability, their mean loss function, statistics corresponding to their loss differentials (they are the number of the models less one), and orders these statistics. It also prints the p-value of the test and the number of eliminated models. If no models with outstanding predictive ability were found, the function prints such an information.

Examples

```r
data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
m <- MDM.test(realized=ts, evaluated=forecasts, q=10, statistic="S", loss.type="AE")
print(m)
```

---

**TB_MA** Checks for a Lag in VMA Process with Tiao-Box Procedure.

Description

This function helps to find a lag in stationary VMA process with Tiao-Box procedure, i.e., the lag length beyond which we are willing to assume that the autocorrelation is essentially zero.

Usage

```r
TB_MA(d, q.max)
```

Arguments

- \( d \) matrix of time-series, assumed to be the stationary VARMA type, columns correspond to time index, and rows to different time-series
- \( q.\max \) numeric indicating the maximum number of lag to be considered

Details

The function searches for correlations smaller than \(-2n^{-0.5}\) or higher than \(2n^{-0.5}\), where \( n \) is the length of the time-series.

Value

numeric indicating the found lag length
References


Examples

data(MDMforecasts)
ts <- MDMforecasts$ts
forecasts <- MDMforecasts$forecasts
l <- loss(realized=ts, evaluated=forecasts, loss.type="SE")
d <- d_t(l)
TB_MA(d=d, q.max=10)
Index

d_t. 3
DM.test, 2

list, 2, 5–7
logical, 2
loss, 4

matrix, 3–8, 10
MDM.selection, 5, 9
MDM.test, 5, 6
MDMforecasts, 7

numeric, 2, 4–6, 10
oilforecasts, 8
print.MDM, 9
TB_MA, 10

vector, 2.4–7