Package: PointFore (via r-universe)

October 26, 2024

Type Package

Title Interpretation of Point Forecasts as State-Dependent Quantiles and Expectiles

Version 0.2.0.9000

Description Estimate specification models for the state-dependent level of an optimal quantile/expectile forecast. Wald Tests and the test of overidentifying restrictions are implemented. Plotting of the estimated specification model is possible. The package contains two data sets with forecasts and realizations: the daily accumulated precipitation at London, UK from the high-resolution model of the European Centre for Medium-Range Weather Forecasts (ECMWF, <https://www.ecmwf.int/>) and GDP growth Greenbook data by the US Federal Reserve. See Schmidt, Katzfuss and Gneiting (2015) <arXiv:1506.01917> for more details on the identification and estimation of a directive behind a point forecast.

License CC0

Imports gmm, boot, car, ggplot2, MASS, stats, lubridate, sandwich

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat, spelling

VignetteBuilder knitr

Language en-US

Repository https://schmidtpk.r-universe.dev

RemoteUrl https://github.com/schmidtpk/pointfore

RemoteRef HEAD

RemoteSha c10c5cba01c4a53fc426158197b9181e463a39b2

Contents

`constant` *Constant specification model*

Description

All specification models can be used as parameter in `estimate.functional`. Specification models are used to denote the quantile or expectile level (depending on the identification function). The constant specification model returns the parameter theta irrespective of the state variable. If theta is not in the unit interval, the constant specification model returns 0 or 1 (depending on which is closer).

Usage

```
constant(stateVariable, theta, ...)
```

Arguments

`stateVariable`	state variable
`theta`	parameter
`...`	...

Value

numeric level

See Also

Other specification models: `logistic_linear()`, `probit_break()`, `probit_linear()`, `probit_spline2()`, `probit_spline3()`

Examples

```
# the returned level does not depend on the state variable
constant(0,.5)
constant(1,.5)

# if theta is not in the unit interval, the constant specification model forces it to be so
constant(0, 2)
constant(0, -1)
```

`estimate.functional` *Estimate Functional*

Description

Estimates the parameter in a specification model for state-dependent quantile or expectile forecasts. For additional detail see the vignettes of the PointFore package.

Usage

```
estimate.functional(
  iden.fct = quantiles,
  model = constant,
  theta0 = NULL,
  Y,
  X,
  stateVariable = NULL,
  other_data = NULL,
  instruments = c("X", "lag(Y)"),
  prewhite = F,
  kernel = "Bartlett",
  bw = bwNeweyWest1987,
  ...
)
```

Arguments

`iden.fct`	identification function. Standard choice is `quantiles`. The alternative is `expectiles`.
`model`	specification model. See `constant` for the simplest example and further suggestions.
`theta0`	starting value for optimization
`Y`	realized values
`X`	forecasts
`stateVariable`	state variable(s) as vector or matrix of column vectors.
`other_data`	optional for construction of instruments

`instruments` instruments (list of character describing instruments or matrix of actual instruments). Use "const" for just the constant as instrument. Standard ist c("X","lag(Y)"), which uses the constant, the forecast and the lagged value of the outcome.

`prewhite` logical or integer. Should the estimating functions be prewhitened? Standard is FALSE. If TRUE or greater than 0 a VAR model of order as.integer(prewhite) is fitted. (see ?gmm)

`kernel` choose kernel for HAC-covariance estimation (see ?gmm). Standard is "Bartlett" Kernel as proposed in Newey and West (1987).

`bw` function describing bandwidth selection (see ?gmm for alternatives). Standard is that the bandwidth depends on the sample length $T$ by $m(T)=T^1/5$.

`...` other parameters for gmm function (see ?gmm)

Value

Object of type `pointfore`. Use `summary` and `plot` methods to illustrate results.

Examples

```
# estimate constant quantile level of GDP forecast
res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
model=constant)
summary(res)
plot(res)

# estimate constant quantile level with only the constant as instrument
res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
model=constant, instruments="const")
summary(res)

## Not run:
# estimate constant expectile level
res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
model=constant, instruments="const", iden.fct = expectiles)
summary(res)
plot(res)

# estimate state-dependent quantile level with linear probit specification model
res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
stateVariable = GDP$forecast, model = probit_linear)
summary(res)
plot(res)

## End(Not run)
```

expectiles *Identification function for state-dependent expectiles*

Description

Identification function for state-dependent expectiles

Usage

```
expectiles(x, y, stateVariable, theta, model, ...)
```

Arguments

`x`	forecast
`y`	realization
`stateVariable`	state variable
`theta`	model parameter to be estimated
`model`	model function
`...`	...

See Also

Other identification functions: `quantiles()`

Examples

```
## Estimate expectile level for constant specification model with estimate.functional

res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
model=constant,
instruments="const",
iden.fct = expectiles)
summary(res)
plot(res)
```

GDP *real GDP realized values and one quarter ahead Greenbook forecasts (1969-2012)*

Description

A dataset containing real GDP growth rate in the United States and according one quarter ahead point forecasts from Federal Reserve's Greenbook. The forecasts were selected to be closest to the middle of the respective quarter among the published Greenbook forecasts of one quarter. The forecasts issued latest in the respective quarter are also given under forecast_late.

Usage

```
GDP
```

Format

A data frame with 176 rows and 2 variables:

- observation: realized GDP growth rate in percentage measured at first vintage (-10.4 – 11.2)
- observation_second: realized GDP growth rate in percentage measured at second vintage (-11.3 – 13.2)
- observation_recent: realized GDP growth rate in percentage measured at most recent vintage (-8.2 – 16.5)
- forecast: according point forecast issued one quarter before (-4.7 – 8.5)
- forecast_late: according point forecast issued latest in the quarter before (-4.7 – 7.9)

Source

https://www.philadelphiafed.org/research-and-data/real-time-center/greenbook-data/philadelphia-data-set

lag *Lagging variables for use in estimate functional*

Description

Lagging variables for use in estimate functional

Usage

```
lag(vector, lag = 1)
```

Arguments

vector	vector to be lagged
lag	number of lags

Value

lagged vector of same length with NAs at beginning

Examples

```
#lag example vector by one lag
lag(c(1,2,3))

#lag example vector by two lags
lag(c(1,2,3,4),lag=2)
```

`logistic_linear` *linear logistic specification model*

Description

All specification models can be used as parameter in `estimate.functional`. Specification models are used to denote the quantile or expectile level (depending on the identification function). The linear logistic specification model depends linear on the state variable with a logistic link function.

Usage

```
logistic_linear(stateVariable, theta, ...)
```

Arguments

`stateVariable`	state variable
`theta`	parameter
`...`	...

Value

numeric level

See Also

Other specification models: `constant()`, `probit_break()`, `probit_linear()`, `probit_spline2()`, `probit_spline3()`

Examples

```
# plot linear logistic specification model with constant quantile/expectile level
plot(function(x) logistic_linear(x,theta=c(0,0)), xlim=c(-1,1))


# plot linear logistic specification model with state-dependent quantile/expectile level
plot(function(x) logistic_linear(x,theta=c(0,5)), xlim=c(-1,1))
```

plot.pointfore *Plots object of class "pointfore"*

Description

Plots object of class "pointfore"

Usage

```
## S3 method for class 'pointfore'
plot(
  x,
  conf.levels = c(0.6, 0.9),
  pdf = TRUE,
  hline = TRUE,
  adjust.factor = 1,
  limits = NULL,
  ...
)
```

Arguments

x	object of class "pointfore"
conf.levels	one or two confidence levels for pointwise confidence intervals
pdf	logic if pdf estimate should be plotted
hline	if TRUE plots horizontal line at 0.5. if numeric plot horizontal line at value.
adjust.factor	adjust factor for estimating pdf (controls smoothness)
limits	2-dimensional vector defining range of x-axis
...	other parameters

Value

plot

Examples

```
#estimate linear probit specification model for quantiles on GDP forecast
res <- estimate.functional(Y=GDP$observation,X=GDP$forecast,
model=probit_linear, stateVariable = GDP$forecast)
#plot results
plot(res)
```

PointFore *PointFore: A package for estimating state-dependent quantile and expectile levels from a time series of point forecasts and observations*

Description

Estimate specification models for the state-dependent level of an optimal quantile/expectile forecast. Wald Tests and the test of overidentifying restrictions are implemented. Plotting of the estimated specification model is possible. The package contains daily accumulated precipitation at London, UK from the high-resolution model of the European Centre for Medium-Range Weather Forecasts (ECMWF, https://www.ecmwf.int/). The package further contains quarterly GDP growth data with observations and forecasts from the Federal Reserve's Greenbook. Based on "Interpretation of Point Forecasts" by Patrick Schmidt, Matthias Katzfuss, and Tilmann Gneiting.

PointFore functions

The main function is `estimate.functional`. It returns an object which can be analyzed with `plot.pointfore` and `summary.pointfore`.

precipitation *daily accumulated precipitation (in millimeter) at London, UK and 24-hours-ahead forecasts of the ECMWF (1969-2011)*

Description

24-hour ahead forecasts of daily accumulated precipitation at London, UK from the high-resolution model of the European Centre for Medium-Range Weather Forecasts (ECMWF). The data contains observations from 2012 to 2016.

Usage

```
precipitation
```

Format

A data frame with 2192 rows and 2 variables:

- Y: daily accumulated precipitation in millimeter at London, UK (0 – 45)
- X: according point forecast in millimeter issued one day ahead (0 – 40)

Details

We thank the ECMWF for their support. Further details can be found on `https://www.ecmwf.int/`.

probit_break *probit break specification model with probit link*

Description

All specification models can be used as parameter in estimate.functional. Specification models are used to denote the quantile or expectile level (depending on the identification function). The probit break specification model depends has a break at zero and a constant level above and below. It applies the probit link function.

Usage

```
probit_break(stateVariable, theta, ...)
```

Arguments

stateVariable	state variable
theta	parameter
...	...

Value

numeric level

See Also

Other specification models: constant(), logistic_linear(), probit_linear(), probit_spline2(), probit_spline3()

Examples

```
# plot break probit specification model with constant quantile/expectile level
plot(function(x) probit_break(x,theta=c(0,0)), xlim=c(-1,1))


# plot linear break specification model with state-dependent quantile/expectile level
plot(function(x) probit_break(x,theta=c(0,5)), xlim=c(-1,1))
```

probit_linear *linear specification model with probit link*

Description

All specification models can be used as parameter in estimate.functional. Specification models are used to denote the quantile or expectile level (depending on the identification function). The linear probit specification model depends linear on the state variable with a probit link function.

Usage

```
probit_linear(stateVariable, theta, ...)
```

Arguments

stateVariable	state variable
theta	parameter
...	other parameters

Value

numeric level

See Also

Other specification models: constant(), logistic_linear(), probit_break(), probit_spline2(), probit_spline3()

Examples

```
# plot linear probit specification model with constant quantile/expectile level
plot(function(x) probit_linear(x,theta=c(0,0)), xlim=c(-1,1))


# plot linear probit specification model with state-dependent quantile/expectile level
plot(function(x) probit_linear(x,theta=c(0,5)), xlim=c(-1,1))
```

probit_spline2 *quadratic spline specification model with probit link*

Description

All specification models can be used as parameter in estimate.functional. Specification models are used to denote the quantile or expectile level (depending on the identifciation function). This specification model depends through a quadratic spline on the state variable and applies a probit link function.

Usage

```
probit_spline2(stateVariable, theta, ...)
```

Arguments

stateVariable	state variable
theta	parameter
...	...

Value

numeric level

See Also

Other specification models: constant(), logistic_linear(), probit_break(), probit_linear(), probit_spline3()

Examples

```
# plot example of quadratic spline specification model with state-dependent quantile/expectile level
plot(function(x) probit_spline2(x,theta=c(0,1,-1)), xlim=c(-2,2))
```

probit_spline3 *cubic spline specification model with probit link*

Description

All specification models can be used as parameter in estimate.functional. Specification models are used to denote the quantile or expectile level (depending on the identification function). This specification model depends through a cubic spline on the state variable and applies a probit link function.

Usage

```
probit_spline3(stateVariable, theta, ...)
```

Arguments

`stateVariable`	state variable
`theta`	parameter
`...`	...

Value

numeric level

See Also

Other specification models: `constant()`, `logistic_linear()`, `probit_break()`, `probit_linear()`, `probit_spline2()`

Examples

```
# plot example of cubic spline specification model with state-dependent quantile/expectile level
plot(function(x) probit_spline3(x,theta=c(0,1,1,-1)), xlim=c(-2,2))
```

`quantiles` *Identification function for state-dependent quantiles*

Description

Main alternative to estimating state-dependent quantiles based on the quantile identification function are state-dependent `expectiles`.

Usage

```
quantiles(x, y, stateVariable, theta, model, ...)
```

Arguments

`x`	forecast
`y`	realization
`stateVariable`	state variable
`theta`	model parameter to be estimated
`model`	model function
`...`	...

See Also

Other identification functions: expectiles()

Examples

```
### estimate expectation of identification function for quantile forecasts

set.seed(1)
y <- rnorm(1000)
x <- qnorm(0.6)
# expectation of identification with quantile level 0.6 is zero
mean(quantiles(x,y,0,0.6,constant))
# expectation of identification function with different quantile level
# (0.5 is the median) is not zero
mean(quantiles(x,y,0,0.5, constant))
```

---

summary.pointfore *Method for object of class pointfore*

Description

It presents results from the estimate.functional estimation as summary does for the lm or gmm class objects for example. It also computes the test of overidentifying restrictions.

Usage

```
## S3 method for class 'pointfore'
summary(object, ...)
```

Arguments

object	An object of class pointfore
...	Other arguments when summary is applied to another class object

Value

It returns a list with the parameter estimates and their standard deviations, t-stat and p-values. It also returns the J-test and p-value for the null hypothesis that the forecast is generated by the postulated functional with an information set that contains the instruments.

Examples

```
# estimate.functional generates a pointfore object...
res <- estimate.functional(Y=GDP$observation, X=GDP$forecast,
model=constant,
instruments="const")

# ...which can be summarized with the \code{summary} function.
summary(res)
```

Index