Package ‘tsPI’

March 17, 2016

Title Improved Prediction Intervals for ARIMA Processes and Structural Time Series

Version 1.0.1

Date 2016-03-17

Author Jouni Helske

Maintainer Jouni Helske <jouni.helske@jyu.fi>

Imports KFAS

Suggests testthat

Description Prediction intervals for ARIMA and structural time series models using importance sampling approach with uninformative priors for model parameters, leading to more accurate coverage probabilities in frequentist sense. Instead of sampling the future observations and hidden states of the state space representation of the model, only model parameters are sampled, and the method is based solving the equations corresponding to the conditional coverage probability of the prediction intervals. This makes method relatively fast compared to for example MCMC methods, and standard errors of prediction limits can also be computed straightforwardly.

License GPL-3

NeedsCompilation yes

Encoding UTF-8

BugReports https://github.com/helske/tsPI/issues

RoxygenNote 5.0.1

Repository CRAN

Date/Publication 2016-03-17 13:14:01

R topics documented:

acv_arma 2
arima_pi 3
avg_coverage_arima 4
avg_coverage_struct 5

dacv_arma 7
information_arma 8
jeffreys 8
struct_pi 9
tsPI 11

Index **12**

`acv_arma` *Compute a theoretical autocovariance function of ARMA process*

Description

Function `acv_arma` computes a theoretical autocovariance function of ARMA process.

Usage

```
acv_arma(phi, theta, n)
```

Arguments

`phi`	vector containing the AR parameters
`theta`	vector containing the MA parameters
`n`	length of the time series

Value

vector of length n containing the autocovariances

See Also

`dacv_arma`.

Examples

```
## Example from Brockwell & Davis (1991, page 92-94)
## also in help page of ARMAacf (from stats)
n <- 0:9
answer <- 2^(-n) * (32/3 + 8 * n) /(32/3)
acv <- acv_arma(c(1.0, -0.25), 1.0, 10)
all.equal(acv/acv[1], answer)
```

`arima_pi` *Prediction Intervals for ARIMA Processes with Exogenous Variables Using Importance Sampling*

Description

Function `arima_pi` computes prediction intervals for ARIMA processes with exogenous variables using importance sampling. For regression coefficients, diffuse (uninformative) prior is used, whereas multiple options for prior distributions for ARMA coefficients are supported.

Usage

```
arima_pi(x, order, xreg = NULL, n_ahead = 1, level = 0.95,
  median = TRUE, se_limits = TRUE, prior = "uniform", custom_prior,
  custom_prior_args = NULL, nsim = 1000, invertibility = FALSE,
  last_only = FALSE, return_weights = FALSE, ...)
```

Arguments

`x`	vector containing the time series
`order`	vector of length 3 with values p,d,q corresponding to the number of AR parameters, degree of differencing and number of MA parameters.
`xreg`	matrix or data frame containing the exogenous variables (not including the intercept which is always included for non-differenced series)
`n_ahead`	length of the forecast horizon.
`level`	desired frequentist coverage probability of the prediction intervals.
`median`	compute the median of the prediction interval.
`se_limits`	compute the standard errors of the prediction interval limits.
`prior`	prior to be used in importance sampling for AR and MA parameters. Defaults to uniform prior. Several Jeffreys' priors are also available (see `jeffreys` for details). If "custom", a user-defined custom prior is used (see next arguments). All priors assume that the ARMA parameters lie in stationarity/invertibility region.
`custom_prior`	function for computing custom prior. First argument must be a vector containing the AR and MA parameters (in that order).
`custom_prior_args`	list containing additional arguments to `custom_prior`.
`nsim`	number of simulations used in importance sampling. Default is 1000.
`invertibility`	Logical, should the priors include invertibility constraint? Default is `FALSE`.
`last_only`	compute the prediction intervals only for the last prediction step.
`return_weights`	Return (scaled) weights used in importance sampling.
`...`	additional arguments for `arima`.

Value

a list containing the prediction intervals. @references

1. Helske, J. and Nyblom, J. (2015). Improved frequentist prediction intervals for autoregressive models by simulation. In Siem Jan Koopman and Neil Shephard, editors, Unobserved Components and Time Series Econometrics. Oxford University Press. http://urn.fi/URN:NBN:fi:jyu-201603141839
2. Helske, J. and Nyblom, J. (2014). Improved frequentist prediction intervals for ARMA models by simulation. In Johan Knif and Bernd Pape, editors, Contributions to Mathematics, Statistics, Econometrics, and Finance: essays in honour of professor Seppo Pynnönen, number 296 in Acta Wasaensia, pages 71–86. University of Vaasa. http://urn.fi/URN:NBN:fi:jyu-201603141836

See Also

tsPI, struct_pi

Examples

```
set.seed(123)
x <- arima.sim(n = 30, model = list(ar = 0.9))

pred_arima <- predict(arima(x, order = c(1,0,0)), n.ahead = 10, se.fit = TRUE)
pred_arima <- cbind(pred = pred_arima$pred,
  lwr = pred_arima$pred - qnorm(0.975)*pred_arima$se,
  upr = pred_arima$pred + qnorm(0.975)*pred_arima$se)

pred <- arima_pi(x, order = c(1,0,0), n_ahead = 10)

ts.plot(ts.union(x,pred_arima, pred[,1:3]), col = c(1,2,2,2,3,3,3),
  lty = c(1,1,2,2,1,2,2))
```

---

avg_coverage_arima — *Compute the average coverage of the prediction intervals computed by naive plug-in method and arima_pi*

Description

Computes expected coverage probabilities of the prediction intervals of ARMA process by simulating time series from the known model.

Usage

```
avg_coverage_arima(phi = NULL, theta = NULL, d = 0, n, n_ahead = 1,
  nsim2, nsim = 100, level = 0.95, prior = "uniform",
  return_all_coverages = FALSE, ...)
```

Arguments

phi	vector containing the AR parameters
theta	vector containing the MA parameters
d	degree of differencing
n	length of the time series
n_ahead	length of the forecast horizon
nsim2	number of simulations used in computing the expected coverage
nsim	number of simulations used in importance sampling
level	desired coverage probability of the prediction intervals
prior	prior to be used in importance sampling. Multiple choices are allowed.
return_all_coverages	return raw results i.e. coverages for each simulations. When FALSE (default), summary statistics are returned.
...	additional arguments to arima_pi.

Value

a list containing the coverage probabilities

See Also

arima_pi.

Examples

```
## Not run:
set.seed(123)
# takes a while, notice se, increase nsim2 to get more accurate results
avg_coverage_arima(phi = 0.9, n = 50, n_ahead = 10, nsim2 = 100)

avg_coverage_arima(phi = 0.9, theta = -0.6, n = 50, n_ahead = 10, nsim2 = 100)

## End(Not run)
```

---

avg_coverage_struct — *Compute the average coverage of the prediction intervals computed by struct_pi and plug-in method*

Description

Computes expected coverage probabilities of the prediction intervals of structural time series model. Note that for the plug-in method only standard deviations are assumed to be identical to their estimates, but the initial values for the states are still treated as diffuse. Because of this, plug-in method often performs relatively well in case of structural time series models compared to similar type of ARIMA models (local level and local linear trend models are closely related to ARIMA(0,1,1) and ARIMA(0,2,2) models), and in some cases even outperforms the importance sampling approach with uniform prior (see examples). This is not suprising, as local level and local linear trend models are closely related to ARIMA(0,1,1) and ARIMA(0,2,2) models, and the effect of uncertainty in MA components is not as significant as the uncertainty of AR components

Usage

```
avg_coverage_struct(type = c("level", "trend", "BSM"), sds, frequency = 1,
  n, n_ahead = 1, nsim2, nsim = 100, level = 0.95, prior = "uniform",
  return_all_coverages = FALSE, ...)
```

Arguments

`type`	Type of model. See `struct_pi`.
`sds`	vector containing the standard deviations of the model (observation error, level, slope, and seasonal).
`frequency`	frequency of the series, needed for seasonal component.
`n`	length of the time series
`n_ahead`	length of the forecast horizon
`nsim2`	number of simulations used in computing the expected coverage
`nsim`	number of simulations used in importance sampling
`level`	desired coverage probability of the prediction intervals
`prior`	prior to be used in importance sampling.
`return_all_coverages`	return raw results i.e. coverages for each simulations. When `FALSE` (default), summary statistics are returned.
`...`	additional arguments to `struct_pi`.

Value

a list containing the coverage probabilities

See Also

`struct_pi`.

Examples

```
## Not run:
set.seed(123)
# takes a while, notice se, increase nsim2 to get more accurate results
avg_coverage_struct(type = "level", sds = c(1, 0.1), n = 50, n_ahead = 10, nsim2 = 100)
avg_coverage_struct(type = "BSM", sds = c(1, 1, 0.1, 10),
 frequency = 4, n = 50, n_ahead = 10, nsim2 = 100)

## End(Not run)
```

---

dacv_arma *Compute the partial derivatives of theoretical autocovariance function of ARMA process*

Description

Function dacv_arma computes the partial derivatives of theoretical autocovariance function of ARMA process

Usage

```
dacv_arma(phi, theta, n)
```

Arguments

phi	vector containing the AR parameters
theta	vector containing the MA parameters
n	length of the time series

Value

matrix containing the partial derivatives autocovariances, each column corresponding to one parameter of vector (phi,theta) (in that order)

See Also

acv_arma.

`information_arma` *Large Sample Approximation of Information Matrix for ARMA process*

Description

Fortran implementation of `InformationMatrixARMA` function of `FitARMA` package, except that the function uses the same ARMA model definition as `arima`, where both the AR and MA parts of the model are on the right side of the equation, i.e. MA coefficients differ in sign compared to `InformationMatrixARMA`.

Usage

```
information_arma(phi = NULL, theta = NULL)
```

Arguments

`phi`	Autoregressive coefficients.
`theta`	Moving average coefficients.

Value

Large sample approximation of information matrix for ARMA process.

References

1. Box, G. and Jenkins, G. (1970). Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.
2. McLeod, A. I. and Zhang, Y., (2007). Faster ARMA maximum likelihood estimation Computational Statistics & Data Analysis 52(4) URL http://dx.doi.org/10.1016/j.csda.2007.07.020

`jeffreys` *Compute different types of importance weights based on Jeffreys's prior*

Description

These functions compute different types of importance weights based on Jeffreys's priors used in `arima_pi`.

Usage

```
approx_joint_jeffreys(psi, xreg = NULL, p, q, n)

approx_marginal_jeffreys(psi, p, q)

exact_joint_jeffreys(psi, xreg = NULL, p, q, n)

exact_marginal_jeffreys(psi, p, q, n)
```

Arguments

psi	vector containing the ar and ma parameters (in that order).
xreg	matrix or data frame containing the exogenous variables (not including the intercept which is always included for non-differenced series)
p	number of ar parameters
q	number of ma parameters
n	length of the time series

See Also

arima_pi.

---

struct_pi *Prediction Intervals for Structural Time Series with Exogenous Variables Using Importance Sampling*

Description

Function struct_pi computes prediction intervals for structural time series with exogenous variables using importance sampling.

Usage

```
struct_pi(x, type = c("level", "trend", "BSM"), xreg = NULL, n_ahead = 1,
  level = 0.95, median = TRUE, se_limits = TRUE, prior = "uniform",
  custom_prior, custom_prior_args = NULL, nsim = 1000, inits = NULL,
  last_only = FALSE, return_weights = FALSE)
```

Arguments

x	vector containing the time series
type	type of model. Possible options are "level", "trend" and "BSM", corresponding to local level, local linear trend, and local linear trend model with seasonal component.

xreg	matrix or data frame containing the exogenous variables (not including the intercept which is always included for non-differenced series)
n_ahead	length of the forecast horizon.
level	desired frequentist coverage probability of the prediction intervals.
median	compute the median of the prediction interval.
se_limits	compute the standard errors of the prediction interval limits.
prior	prior to be used in importance sampling for log-sd parameters. Defaults to uniform prior on logarithm of standard deviations (with constraints that all variances are smaller than 1e7). If "custom", a user-defined custom prior is used (see next arguments).
custom_prior	function for computing custom prior. First argument must be a vector containing the log-variance parameters (observation error, level, slope, and seasonal).
custom_prior_args	list containing additional arguments to custom_prior.
nsim	number of simulations used in importance sampling. Default is 1000.
inits	initial values for log-sds
last_only	compute the prediction intervals only for the last prediction step.
return_weights	Return (scaled) weights used in importance sampling.

Value

a list containing the prediction intervals.

See Also

tsPI, arima_pi @references

1. Helske, J. (2015). Prediction and interpolation of time series by state space models. University of Jyväskylä. PhD thesis, Report 152. http://urn.fi/URN:NBN:fi:jyu-201603111829

Examples

```
pred_StructTS <- predict(StructTS(Nile, type ="level"), n.ahead = 10, se.fit = TRUE)
pred_StructTS <- cbind(pred = pred_StructTS$pred,
  lwr = pred_StructTS$pred - qnorm(0.975)*pred_StructTS$se,
 upr = pred_StructTS$pred + qnorm(0.975)*pred_StructTS$se)

set.seed(123)
pred <- struct_pi(Nile, type = "level", n_ahead = 10)

ts.plot(ts.union(Nile,pred_StructTS, pred[,1:3]), col = c(1,2,2,2,3,3,3),
  lty = c(1,1,2,2,1,2,2))
```

tsPI *Improved Prediction Intervals for ARIMA Processes and Structural Time Series*

Description

Package tsPI computes prediction intervals for ARIMA and structural time series models by using importance sampling approach with uninformative priors for model parameters, leading to more accurate coverage probabilities in frequentist sense. Instead of sampling the future observations and hidden states of the state space representation of the model, only model parameters are sampled, and the method is based solving the equations corresponding to the conditional coverage probability of the prediction intervals. This makes method relatively fast compared to for example MCMC methods, and standard errors of prediction limits can also be computed straightforwardly.

References

1. Helske, J. and Nyblom, J. (2013). Improved frequentist prediction intervals for autoregressive models by simulation. In Siem Jan Koopman and Neil Shephard, editors, Unobserved Components and Time Series Econometrics. Oxford University Press. In press.
2. Helske, J. and Nyblom, J. (2014). Improved frequentist prediction intervals for ARMA models by simulation. In Johan Knif and Bernd Pape, editors, Contributions to Mathematics, Statistics, Econometrics, and Finance: essays in honour of professor Seppo Pynnönen, number 296 in Acta Wasaensia, pages 71–86. University of Vaasa.
3. Helske, J. (2015). Prediction and interpolation of time series by state space models. University of Jyväskylä. PhD thesis, Report 152.

Index

acv_arma, 2, *7*
approx_joint_jeffreys *(*jeffreys*)*, 8
approx_marginal_jeffreys *(*jeffreys*)*, 8
arima, *3*
arima_pi, 3, *4, 5*, *8–10*
avg_coverage_arima, 4
avg_coverage_struct, 5

dacv_arma, *2*, 7

exact_joint_jeffreys *(*jeffreys*)*, 8
exact_marginal_jeffreys *(*jeffreys*)*, 8

information_arma, 8

jeffreys, *3*, 8

struct_pi, *4–6*, 9

tsPI, *4*, *10*, 11
tsPI-package *(*tsPI*)*, 11