Package ‘srlTS’

December 14, 2023

Type Package

Title Sparsity-Ranked Lasso for Time Series

Version 0.1.1

Description An implementation of sparsity-ranked lasso for time series data. This methodology is especially useful for large time series with exogenous features and/or complex seasonality. Originally described in Peterson and Cavanaugh (2022) <doi:10.1007/s10182-021-00431-7> in the context of variable selection with interactions and/or polynomials, ranked sparsity is a philosophy with methods useful for variable selection in the presence of prior informational asymmetry. This situation exists for time series data with complex seasonality, as shown in Peterson and Cavanaugh (2023+) <doi:10.48550/arXiv.2211.01492>, which also describes this package in greater detail. The Sparsity-Ranked Lasso (SRL) for Time Series implemented in 'srlTS' can fit large/complex/high-frequency time series quickly, even with a high-dimensional exogenous feature set. The SRL is considerably faster than its competitors, while often producing more accurate predictions. Also included is a long hourly series of arrivals into the University of Iowa Emergency Department with concurrent local temperature.

Suggests covr, kableExtra, knitr, magrittr, rmarkdown, testthat (>= 3.0.0)

Imports dplyr, methods, ncvreg, RcppRoll, rlang, yardstick

Depends R (>= 3.5)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Config/testthat/edition 3

VignetteBuilder knitr

URL https://petersonr.github.io/srlTS/, https://github.com/petersonR/srlTS/

BugReports https://github.com/petersonR/srlTS/issues

Date 2023-12-13

NeedsCompilation no

Author Ryan Andrew Peterson [aut, cre, cph] (<https://orcid.org/0000-0002-4650-5798>)

Maintainer Ryan Andrew Peterson <ryan.a.peterson@cuanschutz.edu>

Repository CRAN

Date/Publication 2023-12-14 08:00:02 UTC

R topics documented:

`AICc` *internal AICc function for lasso models*

Description

internal AICc function for lasso models

Internal function for obtaining oos results

Internal function for converting time series into model matrix of lags

Usage

```
AICc(fit, eps = 1)

get_oos_results(fits, ytest, Xtest)

get_model_matrix(y, X = NULL, n_lags_max)
```

Arguments

`fit`	an object with logLik method,
`eps`	minimum df used in computation
`fits`	a list of fits with different tuning parameters
`ytest`	validation data
`Xtest`	new X data, including lags
`y`	time series vector
`X`	Additional exogenous features
`n_lags_max`	Maximum number of lags to add

`predict.srlTS` *Predict function for srlTS object*

Description

Predict function for srlTS object

Usage

```
## S3 method for class 'srlTS'
predict(object, n_ahead = 1, X_test, y_test, cumulative = 0, ...)
```

Arguments

`object`	an srlTS object
`n_ahead`	number of times ahead to predict by iteration
`X_test`	a matrix exogenous features
`y_test`	the test series; needed for future predictions (optional; see details)
`cumulative`	should cumulative (rolling) sums be returned (integer indicating number of times to sum)
`...`	currently unused

Details

The ‘y_test‘ argument must be supplied if forecasts are desired or if ‘n_ahead‘ < ‘nrow(X_test)‘. This is because in order to obtain 1-step forecast for, say, the 10th observation in the test data set, the 9th observation of ‘y_test‘ is required. The length of ‘y_test‘ will determine how many forecasts to produce. In order to get true forecasts for the first 30 observations after the training set, one must (currently) produce the set of 1-step, 2-step, 3-step, ..., 30-step ahead predictions.

Value

a vector of predictions

Examples

```
data("LakeHuron")
fit_LH <- srlTS(LakeHuron)
predict(fit_LH)
```

srlTS *Perform time series ranked sparsity methods*

Description

Perform time series ranked sparsity methods

Usage

```
srlTS(
  y,
  X = NULL,
  n_lags_max,
  gamma = c(0, 2^(-2:4)),
  ptrain = 0.8,
  pf_eps = 0.01,
  w_endo,
  w_exo,
  ncvreg_args = list(penalty = "lasso", returnX = FALSE, lambda.min = 0.001)
)

## S3 method for class 'srlTS'
plot(x, log.l = TRUE, ...)

## S3 method for class 'srlTS'
coef(object, choose = c("AICc", "BIC", "all"), ...)

## S3 method for class 'srlTS'
print(x, ...)

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

Arguments

y	univariate time series outcome
X	matrix of predictors (no intercept)
n_lags_max	maximum number of lags to consider
gamma	vector of exponent for weights
ptrain	prop. to leave out for test data
pf_eps	penalty factors below this will be set to zero
w_endo	optional pre-specified weights for endogenous terms
w_exo	optional pre-specified weights for exogenous terms (see details)
ncvreg_args	additional args to pass through to ncvreg

`x`	a srlTS object
`log.l`	Should the x-axis (lambda) be logged?
`...`	passed to downstream functions
`object`	a srlTS object
`choose`	which criterion to use for lambda selection (AICc, BIC, or all)

Details

The default weights for exogenous features will be chosen based on a similar approach to the adaptive lasso (using bivariate OLS estimates). For lower dimensional X, it's advised to set `w_exo="unpenalized"`, because this allows for statistical inference on exogenous variable coefficients via the `summary` function.

Value

A list of class `slrTS` with elements

`fits`	a list of lasso fits
`ncvreg_args`	arguments passed to ncvreg
`gamma`	the (negative) exponent on the penalty weights, one for each fit
`n_lags_max`	the maximum number of lags
`y`	the time series
`X`	the utilized matrix of exogenous features
`oos_results`	results on test data using best of fits
`train_idx`	index of observations used in training data

x invisibly

a vector of model coefficients

x (invisibly)

the summary object produced by ncvreg evaluated at the best tuning parameter combination (best AICc).

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

Peterson, R.A., Cavanaugh, J.E. Ranked sparsity: a cogent regularization framework for selecting and estimating feature interactions and polynomials. AStA Adv Stat Anal (2022). https://doi.org/10.1007/s10182-021-00431-7

See Also

predict.srlTS

Examples

```
data("LakeHuron")
fit_LH <- srlTS(LakeHuron)
fit_LH
coef(fit_LH)
plot(fit_LH)
```

`uihc_ed_arrivals` *Hourly arrivals into the University of Iowa Hospital Emergency Department*

Description

A data set containing the 17 columns described below. There are 41640 observations running from 2013 to 2018. Data set are already sorted by time.

Usage

```
uihc_ed_arrivals
```

Format

a data frame with 17 columns and 41640 rows:

Year Calendar year

Quarter Fiscal year quarter

Month Integer for month of year

Day Integer for day of month

Hour Integer for hour of day

Arrivals Number of arrivals into the ED (outcome)

Date Date

Weekday Indicator for day of week

temp hourly concurrent temperature

xmas Christmas day indicator

xmas2 Day after Christmas

nye New Years Eve indicator

nyd New Years Day indicator

thx Thanksgiving day indicator

thx Thanksgiving day (after) indicator

ind Independence day indicator

game_Day Hawkeye football game day indicator

Source

UIHC Emergency Department.

Index