# Package ‘onlineVAR’

May 8, 2019

**Title** Online Fitting of Time-Adaptive Lasso Vector Auto Regression

**Version** 0.1-1

**Date** 2019-05-05

**Depends** R (>= 2.10.0)

**Imports** parallel, lattice

**Suggests**

**Description** Functions for recursive online fitting of time-adaptive lasso vector auto regression. A recursive coordinate descent algorithm is used to estimate sparse vector auto regressive models and exponential forgetting is applied to allow model changes. Details can be found in Jakob W. Messner and Pierre Pinson (2018). ``Online adaptive LASSO estimation in Vector Auto Regressive models for wind power forecasting in high dimension''. International Journal of Forecasting, in press. <doi:10.1016/j.ijforecast.2018.02.001>.

**License** GPL-2 | GPL-3

**NeedsCompilation** yes

**Author** Jakob Messner [aut, cre]

**Maintainer** Jakob Messner <`jakob.messner@posteo.net`>

**Repository** CRAN

**Date/Publication** 2019-05-08 07:50:03 UTC

## R topics documented:

## aemo — *Wind power data from 22 wind farms in south eastern Australia*

### Description

Wind power data from 22 wind farms in south eastern Australia provided by the Australian Electricity Market Operator (AEMO) and pre-processed by Jethro Browell and Stefanos Delikaraoglou. The data set contains data from 2013 in 5 minute resolution and is a subset of the data set provided at http://dx.doi.org/10.15129/9e1d9b96-baa7-4f05-93bd-99c5ae50b141. The columns are the wind farms sorted from west to east. Meta data on the capacity and location of the wind farms can be accessed with `attr(aemo, "meta")`.

### Usage

```
data("aemo")
```

### Format

A data frame with 22 columns for each wind farm and 105120 rows.

### Source

http://dx.doi.org/10.15129/9e1d9b96-baa7-4f05-93bd-99c5ae50b141 https://benjaminjweise.carto.com/tables/aemo_wind_p

### References

Dowell J, Pinson P (2016), Very-Short-Term Probabilistic Wind Power Forecasts by Sparse Vector Autoregression. *IEEE Transactions on Smart Grid*, 7(2), 763-770, DOI: 10.1109/TSG.2015.2424078

## coef.onlineVAR — *Methods for onlineVAR Objects*

### Description

Methods for extracting information from fitted `onlineVAR` objects.

### Usage

```
## S3 method for class 'onlineVAR'
coef(object, time = "last", s = NULL, lags = NULL, ...)
```

### Arguments

| | |
|---|---|
| `object` | An object of class `"onlineVAR"`. |
| `time` | Time step from which coefficient matrix estimates are taken. Default is "last" to take the estimates from the last update. Other time steps can only be specified if model has been fitted with `trace=TRUE`. |
| `s` | Optional index specifying which lambda in the lambda.ratio sequence to use. If not specified, optimum s is taken based on weighted squared error. |
| `lags` | Optional vector specifying for which lags coefficients should be returned. If not specified, coefficient matrices for all lags are returned. |
| `...` | Required for S3 compatibility. |

### Details

In addition to the `coef` method above, also the standard extractor functions `print` and `summary` for `"onlineVAR"` objects are available.

### See Also

`onlineVAR`

---

## `onlineVAR` *Online Fitting of Time-Adaptive Lasso Vector Auto Regression.*

---

### Description

Recursive fitting of time-adaptive lasso vector auto regressive models with an online cyclic coordinate descent algorithm.

### Usage

```
onlineVAR(y, nu = 1, lags = NULL, ahead = NULL, lambda.ratio = NULL,
  burnin = NULL, control = onlineVAR.control(lambda.ratio, ...), ...)

onlineVAR.fit(x, y, nu, lags, ahead, lambda.ratio, fit, burnin,
  control = onlineVAR.control())
```

### Arguments

| | |
|---|---|
| `y` | Vector time series with columns for each variable and rows for each time step. |
| `x` | Lagged time series data. Columns 1 to `ncol(y)` contain lag-1 data, columns `ncol(y)+1` to `2*ncol(y)` lag-2 data and so on. |
| `nu` | Forgetting factor for exponential forgetting. |
| `lags` | Number of considered lags. Order of VAR model. |
| `ahead` | Look ahead time for predictions. Most recent data that are available. The fitted model only considers lags larger than `ahead`. |

| | |
|---|---|
| `lambda.ratio` | Vector of penalization parameters as fractions of the minimum lambda for which all coefficients are zero. If not specified a sequence of lambda values is used. See `onlineVAR.control` for details. |
| `burnin` | Burn-in period, i.e., the number of time steps that are used to stabilize the model fit. This is particularly important for selecting the optimum lambda. Default is the effective training data length `1/(1-nu)` for a vector of `lambda.ratio` and 0 for a single valued `lambda.ratio`. |
| `control` | A list of control parameters. Default is `onlineVAR.control(lambda.ratio)`. |
| `fit` | List of various starting values that are used for the online updating algorithm. |
| `...` | Arguments to be used to form the default `control` argument if it is not supplied directly. |

## Details

`onlineVAR` recursively fits time-adaptive lasso vector auto regressive models with an online coordinate descent algorithm that updates the model fit at each time step. By default these models are fitted for a sequence of lasso penalization parameters and predictions are generated for each time step with the parameter with the smallest weighted mean squared error.

`onlineVAR.fit` is the lower level function where the actual fitting takes place.

## Value

An object of class `"onlineVAR"`, i.e., a list with the following elements.

| | |
|---|---|
| `coef` | List of intercept vector and coefficient matrices for the different lags. |
| `fit` | List of various values that are used for the online updating algorithm. This is mainly used as starting values for further updates. |
| `nu` | Forgetting factor for exponential forgetting. |
| `pred` | Predictions. Matrix of the same size as the input data but with predictions from adaptive lasso VAR. The first `burnin+ahead+lags` rows are filled with `NA` |
| `predall` | If `predall=TRUE` in `control` predictions for all penalization parameters in `lambda.ratio`. Else `NULL` |
| `residuals` | Prediction errors. |
| `lags` | Number of considered lags. Order of VAR model. |
| `ahead` | Look ahead time for predictions. |
| `lambda.ratio` | Sequence of lasso penalization parameters |
| `trace` | If `trace=TRUE` a list of coefficient estimates for each time step. Else `FALSE` |
| `burnin` | Burn-in period, i.e., the number of data that are used to stabilize the model fit. |
| `call` | Function call. |

## References

Messner JW, Pinson P (2016). Online Adaptive LASSO Estimation in Vector Auto Regressive Models for Wind Power Forecasting in High Dimension. *International Journal of Forecasting*, in press. `http://pierrepinson.com/docs/MessnerPinson18.pdf`.

## See Also

predict.onlineVAR, plot.onlineVAR

## Examples

```
data(aemo)

## use only subset of first 8000 time steps
y <- aemo[1:8000,]

## fit online lasso VAR
onlinefit <- onlineVAR(y, nu = 0.99, lags = 1, ahead = 1)

## plot coefficient matrix from last update
plot(onlinefit)

## compare mean root mean squared error to persistence
c(onlinefit   = mean(sqrt(apply((predict(onlinefit)-y)^2, 2,
    mean, na.rm = TRUE))),
  persistence = mean(sqrt(apply((aemo[1000:7999,]-y[1001:8000,])^2, 2,
    mean))))
```

---

## `onlineVAR.control` *Auxiliary Function for Controlling onlineVAR Fitting*

---

### Description

Auxiliary function for `onlineVAR` fitting.

### Usage

```
onlineVAR.control(lambda.ratio = NULL, nlambda = NULL,
  lambda.min.ratio = NULL, abstol = 0.001, trace = FALSE, start = NULL,
  parallel = FALSE, predall = FALSE)
```

### Arguments

| | |
|---|---|
| `lambda.ratio` | Vector of penalization parameters as fractions of the minimum lambda for which all coefficients are zero. If not specified a sequence of lambda values is generated based on `nlambda` and `lambda.min.ratio`. If supplied, `nlambda` and `lambda.min.ratio` are ignored. |
| `nlambda` | Number of lasso penalization parameters lambda. Default is 10. |
| `lambda.min.ratio` | Smallest value of lambda.ratio. Default is 0.0001 |
| `abstol` | Absolute tolerance for coordinate descent convergence. In each time step the algorithm stops when the sum of coefficient estimates does not change more than `abstol`. Default is 0.001. |

| | |
|---|---|
| `trace` | If `TRUE` coefficient estimates are stored for all time steps. If `FALSE` coefficient matrices are only stored for the last time step to save memory. |
| `start` | Object of class `"onlineVAR"`. Coefficient estimates from the last time step are used as staring values. Can be used to continue updating the model with new data. |
| `parallel` | If `TRUE` the model fitting for the different lambda is parallelized. |
| `predall` | Logical whether predictions from all penalization parameters in the sequence are stored. |

## Value

An list of components named as the arguments.

| | |
|---|---|
| `nlambda` | Number of lasso penalization parameters in the lambda sequence. |
| `lambda.min.ratio` | Smallest value for lambda.ratio. |
| `abs.tol` | Absolute tolerance for coordinate descent convergence. |
| `lambda.ratio` | Lambda sequence as fractions of the minimum lambda for which all coefficients are zero. |
| `trace` | Logical whether coefficients should be stored for all time steps. |
| `start` | Starting values. |
| `parallel` | Logical whether the model fitting for the different lambda is parallelized. |
| `predall` | Logical whether prediction from all penalization parameters in the sequence are stored. |

## See Also

`onlineVAR`

---

`plot.onlineVAR` *Coefficient matrix plots for onlineVAR objects.*

---

## Description

Generates plots of estimated coefficient matrices for onlineVAR fits by using the `levelplot` function from `lattice`.

## Usage

```
## S3 method for class 'onlineVAR'
plot(x, lag = 1, time = "last", s = NULL, col = NULL,
  at = NULL, xlab = "", ylab = "", ...)
```

### Arguments

| | |
|---|---|
| `x` | `"onlineVAR"` object, typically output from `onlineVAR`. |
| `lag` | Lag for which coefficient matrix is plotted. |
| `time` | Time step from which coefficient matrix estimates are taken. Default is `"last"` to take the estimates from the last update. Other time steps can only be specified if model has been fitted with `trace=TRUE`. |
| `s` | Optional index specifying which lambda in the lambda.ratio sequence to use. If not specified, optimum s is taken based on weighted squared error. |
| `col` | Color palette. Default is a color palette with reddish colors for negative and blueish colors for positive coefficients. |
| `at` | A numeric vector giving the breakpoints for the color palette. |
| `xlab` | A title for the x axis. |
| `ylab` | A title for the y axis. |
| `...` | Further arguments passed to `levelplot`. |

### See Also

`onlineVAR`, `levelplot`

### Examples

```
data(aemo)

## use only subset of first 8000 time steps
y <- aemo[1:8000,]

## fit online lasso VAR
onlinefit <- onlineVAR(y, nu = 0.99, lags = 1, ahead = 1)

## plot coefficient matrix from last update
plot(onlinefit)
```

---

## `predict.onlineVAR` *Predictions for onlineVAR Models*

---

### Description

Obtains predictions for onlineVAR models.

### Usage

```
## S3 method for class 'onlineVAR'
predict(object, newdata, s = NULL, ...)
```

## Arguments

| | |
|---|---|
| `object` | An object of class `"onlineVAR"`. |
| `newdata` | Optional time series data to predict. If supplied, predictions are derived with coefficients from the last time step that was used for fitting `object` |
| `s` | Optional index specifying which lambda in the lambda.ratio sequence to use for prediction. |
| `...` | Required for S3 compatibility. |

## Details

By default, predictions derived during the online fitting are returned. By specifying `newdata` predictions for a new time series are generated with coefficient estimates from the last time step of the online fitting.

## Value

Time series matrix of predictions of the same size as the input data used for model fitting or, if supplied, of the same size as `newdata`.

## See Also

`onlineVAR`

# Index