Package 'PRISM.forecast'

January 20, 2025

Type Package

Title Penalized Regression with Inferred Seasonality Module - Forecasting Unemployment Initial Claims using 'Google Trends' Data

Version 0.2.1

Maintainer Dingdong Yi <ryan.ddyi@gmail.com>

Description Implements Penalized Regression with Inferred Seasonality Module (PRISM) to generate forecast estimation of weekly unemployment initial claims using 'Google Trends' data. It includes required data and tools for backtesting the performance in 2007-2020.

License GPL-2

Imports xts, glmnet, zoo, stats, utils

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

URL https://github.com/ryanddyi/prism

BugReports https://github.com/ryanddyi/prism/issues

NeedsCompilation no

Author Dingdong Yi [aut, cre],
Samuel Kou [aut],
Shaoyang Ning [aut]

Repository CRAN

Date/Publication 2020-10-21 04:40:05 UTC

Contents

`back_test` *Out-of-sample prediction for whole period*

Description

Out-of-sample prediction for whole period

Usage

```
back_test(
  n.lag = 1:52,
  s.window = 52,
  n.history = 700,
  stl = TRUE,
  n.training = 156,
  UseGoogle = T,
  alpha = 1,
  nPred = 0,
  discount = 0.015,
  sepL1 = F
)
```

Arguments

`n.lag`	the number of lags to be used as regressor in Stage 2 of PRISM (by default = 1:52 for weekly data)
`s.window`	seasonality span in seasonal decomposition (by default = 52 for weekly data)
`n.history`	length of training period (e.g. in weeks) for seasonal decomposition.
`stl`	if TRUE, use STL seasonal decomposition; if FALSE, use classic additive seasonal decomposition.
`n.training`	length of training period in Stage 2, penalized linear regression (by default = 156)
`UseGoogle`	boolean variable indicating whether to use Google Trend data.
`alpha`	penalty between lasso and ridge. alpha=1 represents lasso, alpha=0 represents ridge, alpha=NA represents no penalty (by default alpha = 1).
`nPred`	the number of periods ahead for forecast. nPred = 0,1,2,3.
`discount`	exponential weighting: (1-discount)^lag.
`sepL1`	if TRUE, use separate L1 regularization parameters for time series components and exogenous variables (Goolgle Trend data)

Value

`prediction` `nPred` week ahead prediction of the whole periods (07 - 20).

Examples

```
claim_data = load_claim_data()

# It may take a few minutes.
prism_prediction = back_test()
# evaluate the out-of-sample prediction error as a ratio to naive method
evaluation_table(claim_data, prism_prediction)
```

`evaluation_table` *Out-of-sample prediction evaluation*

Description

Out-of-sample prediction evaluation

Usage

```
evaluation_table(claim_data, prism_prediction)
```

Arguments

`claim_data`	the output of load_claim_data().
`prism_prediction`	the output of back_test().

`load_5y_search_data` *Load Goolge Trends data and initial claims data*

Description

Load weekly unemployment initial claim data and related Google Trend data over 5-year span (each week ends on the Saturday). The list of Google search terms is the same as in paper.

Usage

```
load_5y_search_data(folder = "0408")
```

Arguments

`folder`	foldernames for a certain periods of Google Trends data. The foldernames are "0408", "0610", "0812", "1014", "1216", "1418", "1620". For example, the folder "0408" is for 2004-2008.

Value

A list of following named xts objects

- `claim.data` unemployment initial claim data of the same span as Google Trend data.
- `claim.all` load all unemployment initial claim data since 1967
- `claim.early` unemployment initial claim data from 1980-01-06 to the start of `claim.data`.
- `allSearch` Google Trends data of a span over five years. It is in the scale of 0 – 100.

`load_claim_data` *Load unemployment initial claims data*

Description

Load weekly unemployment initial claim data (each week ends on the Saturday).

Usage

```
load_claim_data(GT.startDate = "2004-01-03", GT.endDate = "2016-12-31")
```

Arguments

`GT.startDate`	start date of claim data
`GT.endDate`	end date of claim data

Value

A list of following named xts objects

- `claim.data` unemployment initial claim data from GT.startDate to GT.endDate.
- `claim.all` load all unemployment initial claim data since 1967
- `claim.early` unemployment initial claim data prior to GT.startDate

prism *PRISM function*

Description

A function for nowcasting and forecasting time series.

Usage

```
prism(
  data,
  data.early,
  GTdata,
  stl = TRUE,
  n.history = 700,
  n.training = 156,
  alpha = 1,
  UseGoogle = T,
  nPred.vec = 0:3,
  discount = 0.015,
  sepL1 = F
)
```

Arguments

data	time series of interest as xts, last element can be NA. (e.g., unemployment initial claim data in the same period as GTdata).
data.early	historical time series of response variable before contemporaneous exogenous data, GTdata is available. (e.g., unemployment initial claim prior to 2004)
GTdata	contemporaneous exogenous data as xts. (e.g., Google Trend data)
stl	if TRUE, use STL seasonal decomposition; if FALSE, use classic additive seasonal decomposition.
n.history	training period for seasonal decomposition. (by default = 700 wks)
n.training	length of regression training period (by default = 156)
alpha	penalty between lasso and ridge. alpha=1 represents lasso, alpha=0 represents ridge, alpha=NA represents no penalty.
UseGoogle	boolean variable indicating whether to use Google Trend data.
nPred.vec	the number of periods ahead for forecast. nPred.vec could be a vector of intergers. e.g. nPred.vec=0:3 gives results from nowcast to 3-week ahead forecast.
discount	exponential weighting: (1-discount)^lag (by default = 0.015).
sepL1	if TRUE, use separate L1 regularization parameters for time series components and exogenous variables (Goolgle Trend data)

Value

A list of following named objects

- `coef` coefficients for Intercept, z.lags, seasonal.lags and exogenous variables.
- `pred` a vector of prediction with `nPred.vec` weeks forward.

Examples

```
prism_data = load_5y_search_data('0610')
data = prism_data$claim.data[1:200] # load claim data from 2006-01-07 to 2009-10-31
data[200] = NA # delete the data for the latest date and try to nowcast it.

data.early = prism_data$claim.earlyData # load claim prior to 2006
GTdata = prism_data$allSearch[1:200] # load Google trend data from 2006-01-07 to 2009-10-31

result = prism(data, data.early, GTdata) # call prism method
result$pred # output 0-3wk forward prediction
```

---

prism_batch *PRISM stage 2 by batch*

Description

PRISM penalized linear regression function for a range of time (only used internally for back testing)

Usage

```
prism_batch(
  data,
  GTdata,
  var,
  n.training = 156,
  UseGoogle = T,
  alpha = 1,
  nPred.vec = 0:3,
  start.date = NULL,
  n.weeks = NULL,
  discount = 0.015,
  sepL1 = F
)
```

Arguments

data	time series of interest as xts, last element can be NA. (e.g., unemployment initial claim data in the same period as GTdata).
GTdata	contemporaneous exogenous data as xts. (e.g., Google Trend data)
var	generated regressors from stage 1.
n.training	length of regression training period (by default = 156)
UseGoogle	boolean variable indicating whether to use Google Trend data.
alpha	penalty between lasso and ridge. alpha=1 represents lasso, alpha=0 represents ridge, alpha=NA represents no penalty.
nPred.vec	the number of periods ahead for forecast. nPred.vec could be a vector of interg-ers. e.g. nPred.vec=0:3 gives results from nowcast to 3-week ahead forecast.
start.date	the starting date for forecast. If NULL, the forecast start at the earliest possible date.
n.weeks	the number of weeks in the batch. If NULL, the forecast end at the latest possible date.
discount	exponential weighting: (1-discount)^lag (by default = 0.015)
sepL1	if TRUE, use separate L1 regularization parameters for time series components and exogenous variables (Goolgle Trend data)

Value

A list of following named objects

- coef coefficients for Intercept, z.lags, seasonal.lags and exogenous variables.
- pred prediction results for n.weeks from start.date.

---

var_generator *PRISM regressors generator*

Description

Stage 1 of PRISM. The function generates prism seasonal components and seasonally adjusted lag components.

Usage

```
var_generator(
  data,
  data.early,
  stl = TRUE,
  n.lag = 1:52,
  s.window = 52,
  n.history = 700
)
```

Arguments

`data`	time series of interest as xts, last element can be NA.
`data.early`	historical time series of response variable before Google Trend data is available. (e.g., unemployment initial claim prior to 2004)
`stl`	if TRUE, use STL seasonal decomposition; if FALSE, use classic additive seasonal decomposition.
`n.lag`	the number of lags to be used as regressor in Stage 2 of PRISM (by default = 1:52 for weekly data)
`s.window`	seasonality span (by default = 52 for weekly data)
`n.history`	training period for seasonal decomposition. (by default = 700 wks)

Value

A list of following named objects

- `y.lags` seasonally adjusted components, z_lag, and seasonal components, s_lag.

Index