Package ‘PRISM.forecast’

July 1, 2018

Type Package
Title Penalized Regression with Inferred Seasonality Module - Forecasting Unemployment Initial Claims using ‘Google Trends’ Data
Version 0.1.6
Maintainer Dingdong Yi <ddyi@fas.harvard.edu>
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 2007-2016.
License GPL-2
Imports xts, glmnet, zoo, stats, utils
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
URL https://github.com/ryanddyi/prism
BugReports https://github.com/ryanddyi/prism/issues
NeedsCompilation no
Author Dingdong Yi [aut, cre], Samuel Kou [aut]
Repository CRAN
Date/Publication 2018-07-01 13:31:10 UTC

R topics documented:

back_test ......................................................... 2
evaluation_table ............................................... 3
load_5y_search_data .......................................... 3
load_claim_data ................................................ 4
prism ............................................................ 4
prism_batch .................................................... 6
var_generator .................................................. 7
Description

Out-of-sample prediction for whole period

Usage

```r
back_test(n.lag = 1:52, s.window = 52, n.history = 700, stl = TRUE,
          n.training = 156, UseGoogle = T, alpha = 1, nPred = 0,
          discount = 0.01, 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 (Google Trend data)

Value

- `prediction nPred` week ahead prediction of the whole periods (07 - 16).

Examples

```r
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 Google 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**: folder names for a certain periods of Google Trends data. The folder names are "0408", "0610", "0812", "1014", "1216". 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 = 0.01, sepL1 = f)
prism

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 integers. e.g. nPred.vec=0:3 gives results from nowcast to 3-week ahead forecast.
discount exponential weighting: (1-discount)^lag (by default = 0.01).
sepL1 if TRUE, use separate L1 regularization parameters for time series components and exogenous variables (Google 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$claim.earlyData # load claim prior to 2006
Gtdata = prism_data$sallSearch[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
Description

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

Usage

```r
prism_batch(data, GTdata, var, n.training = 156, UseGoogle = T, alpha = 1, nPred.vec = 0:3, start.date = NULL, n.weeks = NULL, discount = 0.01, 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 integers. 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 = 0.01)

sepL1 if TRUE, use separate L1 regularization parameters for time series components and exogenous variables (Google 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.
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

back_test, 2
evaluation_table, 3
load_5y_search_data, 3
load_claim_data, 4
prism, 4
prism_batch, 6
var_generator, 7