Package ‘argo’

May 24, 2023

Type Package

Title Accurate Estimation of Influenza Epidemics using Google Search Data

Version 3.0.2

Date 2023-05-23

Author Shaoyang Ning, Shihao Yang, S. C. Kou

Maintainer Shihao Yang <shihao.yang@isye.gatech.edu>

Description Augmented Regression with General Online data (ARGO) for accurate estimation of influenza epidemics in United States on national level, regional level and state level. It replicates the method introduced in paper Yang, S., Santillana, M. and Kou, S.C. (2015) <doi:10.1073/pnas.1515373112>; Ning, S., Yang, S. and Kou, S.C. (2019) <doi:10.1038/s415 019-41559-6>; Yang, S., Ning, S. and Kou, S.C. (2021) <doi:10.1038/s41598-021-83084-5>.

License GPL-2

Imports xts, glmnet, zoo, XML, xtable, Matrix, boot

Suggests testthat

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2023-05-24 11:20:14 UTC

R topics documented:

argo *Construct ARGO object*

Description

Wrapper for ARGO. The real work horse is glmnet package and/or linear model.

Usage

```
argo(
  data,
  exogen = xts::xts(NULL),
  N_lag = 1:52,
  N_training = 104,
  alpha = 1,
  use_all_previous = FALSE,
  mc.cores = 1,
  schedule = list()
)
```

Arguments

`data` response variable as xts, last element can be NA. If the response is later revised, it should be an xts that resembles upper triangular square matrix, with each column being the data available as of date of column name

`exogen` exogenous predictors, default is NULL

`N_lag` vector of the AR model lags used, if NULL then no AR lags will be used

`N_training` number of training points, if `use_all_previous` is true, this is the least number of training points required

`alpha` penalty between lasso and ridge, alpha=1 represents lasso, alpha=0 represents ridge, alpha=NA represents no penalty

`use_all_previous`
boolean variable indicating whether to use "all available data" (when `TRUE`) or "a sliding window" (when `FALSE`) for training

`mc.cores` number of cores to compute argo in parallel

`schedule` list to specify prediction schedule. Default to have `y_gap` as 1, and `forecast` as 0, i.e., nowcasting with past week ILI available from CDC.

Details

This function takes the time series and exogenous variables (optional) as input, and produces out-of-sample prediction for each time point.

Value

A list of following named objects

- `pred` An xts object with the same index as input, which contains historical nowcast estimation
- `coef` A matrix contains historical coefficient values of the predictors.
- `parm` Parameter values passed to argo function.
- `penalfac` the value of lambda ratio selected by cross-validation, NULL if `lamid` is NULL or has only one level.
- `penalregion` the lambda ratios that has a cross validation error within one standard error of minimum cross validation error

References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

Examples

```
GFT_xts <- xts::xts(exp(matrix(rnorm(180), ncol=1)), order.by = Sys.Date() - (180:1))
randomx <- xts::xts(exp(matrix(rnorm(180*100), ncol=100)), order.by = Sys.Date() - (180:1))

argo_result1 <- argo(GFT_xts)
argo_result2 <- argo(GFT_xts, exogen = randomx)
```

`argo2` *ARGO second step*

Description

Wrapper for ARGO second step. Best linear predictor / Bayesian posterior

Usage

```
argo2(truth, argo1.p, argo.nat.p)
```

Arguments

`truth`	prediction target
`argo1.p`	argo first step prediction
`argo.nat.p`	argo national level prediction

References

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

Examples

```
truth <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo1.p <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo.nat.p <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo2result <- argo2(truth, argo1.p, argo.nat.p)
```

argo2_main *main function for argo2*

Description

main function that reproduce the results in ARGO2 paper

Usage

```
argo2_main(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  save.folder = NULL
)
```

Arguments

`gt.folder`	folder with Google Trends files, which should be thousands of csv file such as "US-MA_fever cough.csv" or "US-NY_cold or flu.csv"
`ili.folder`	folder with ILINet data files: "ILINet_nat.csv" and "ILINet_regional.csv"
`population.file`	file path to population csv file
`gft.file`	file path to Google Flu Trends csv file
`save.folder`	output folder to save graphics. If NULL then do not output graphics.

References

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

Examples

```
## Not run:
download.file("https://scholar.harvard.edu/files/syang/files/gt2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")
argo2_main(
  gt.folder=gt.folder,
  ili.folder=system.file("regiondata", "ili20161121", package = "argo"),
  population.file=system.file("regiondata", "Population.csv", package = "argo"),
  gft.file=system.file("regiondata", "GFT.txt", package = "argo")
)

## End(Not run)
```

argox_main *main function for argox*

Description

Main function that reproduce the results in ARGOX paper. The datasets are available at Harvard Dataverse <doi:10.7910/DVN/2IVDGK>.

Usage

```
argox_main(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  mix,
  save.folder = NULL,
  NCORES = 8
)
```

Arguments

gt.folder	folder with Google Trends files, which should be thousands of csv file such as "US-MA_fever cough.csv" or "US-NY_cold or flu.csv"
ili.folder	folder with ILINet data files: "ILINet_nat.csv" and "ILINet_regional.csv"

`population.file`
: file path to population csv file

`gft.file`
: file path to Google Flu Trends csv file

`mix`
: the weighted avarage mixing of raw state-level Google Trends data. Set to be 0 for stand-alone model. Set to be 1/3 for spatial-pooling model.

`save.folder`
: output folder to save graphics. If NULL then do not output graphics.

`NCORES`
: number of parallel cpu cores to be used.

References

Yang, S., Ning, S. & Kou, S.C. Use Internet search data to accurately track state level influenza epidemics. Sci Rep 11, 4023 (2021)

---

argo_main *main function for argo*

Description

main function that reproduce the results in ARGO paper

Usage

```
argo_main(save.folder = NULL)
```

Arguments

`save.folder`
: output folder to save graphics. If NULL then do not output graphics.

Examples

```
argo_main()
```

`bootstrap_relative_efficiency` *bootstrap relative efficiency confidence interval*

Description

This function is used to reproduce the ARGO bootstrap confidence interval

Usage

```
bootstrap_relative_efficiency(
  pred_data,
  model_good,
  model_bench,
  l = 50,
  N = 10000,
  truth = "CDC.data",
  sim = "geom",
  conf = 0.95,
  type = c("mse", "mape", "mae", "mspe", "rmse", "rmspe")
)
```

Arguments

`pred_data` A matrix that contains the truth vector and the predictions. It can be data.frame or xts object

`model_good` The model to evaluate, must be in the column names of pred_data

`model_bench` The model to compare to, must be in the column names of pred_data

`l` stationary bootstrap mean block length

`N` number of bootstrap samples

`truth` the column name of the truth

`sim` simulation method, pass to boot::tsboot

`conf` confidence level

`type` Must be one of "mse" (mean square error), "mape" (mean absolute percentage error), or "mae" (mean absolute error)

Value

A vector of point estimate and corresponding bootstrap confidence interval

Examples

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=5)), order.by = Sys.Date() - (200:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"
bootstrap_relative_efficiency(
  pred_data = GFT_xts,
  model_good = "col2",
  model_bench = "col3",
  truth="CDC.data",
  N = 100
)
```

boot_re *wrapper for bootstrap relative efficiency confidence interval*

Description

This function is used to wrap the `bootstrap_relative_efficiency`, taking vectorized arguments.

Usage

```
boot_re(
  pred_data,
  period.all,
  model_good,
  bench.all,
  type,
  truth = "CDC.data",
  l = 50,
  N = 10000,
  sim = "geom",
  conf = 0.95
)
```

Arguments

`pred_data` A matrix that contains the truth vector and the predictions. It can be data.frame or xts object

`period.all` vector of the periods to evaluate relative efficiency

`model_good` The model to evaluate, must be in the column names of pred_data

`bench.all` vector of the models to compare to, must be in the column names of pred_data

`type` Must be one of "mse" (mean square error), "mape" (mean absolute percentage error), or "mae" (mean absolute error)

`truth` the column name of the truth

`l` stationary bootstrap mean block length

N	number of bootstrap samples
sim	simulation method, pass to boot::tsboot
conf	confidence level

Value

A vector of point estimate and corresponding bootstrap confidence interval

Examples

```
GFT_xts = xts::xts(exp(matrix(rnorm(500), ncol=5)), order.by = Sys.Date() - (100:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"

boot_re(
  pred_data = GFT_xts,
  period.all = c(paste0(zoo::index(GFT_xts)[1], "/", zoo::index(GFT_xts)[50]),
                 paste0(zoo::index(GFT_xts)[51], "/", zoo::index(GFT_xts)[100])),
  model_good = "col2",
  bench.all = c("col3", "col4"),
  type = "mse",
  truth="CDC.data",
  l = 5,
  N = 20
)
```

gt.parser.pub.api *Parsing each Google Trends file downloaded from Google Trends API*

Description

Parsing each Google Trends file downloaded from Google Trends API

Usage

```
gt.parser.pub.api(gt.folder, f)
```

Arguments

gt.folder	folder that contains Google Trends file
f	filename for Google Trends file

gt.parser.pub.web *Parsing each Google Trends file downloaded from website*

Description

Parsing each Google Trends file downloaded from website

Usage

```
gt.parser.pub.web(gt.folder, f)
```

Arguments

gt.folder	folder that contains Google Trends file
f	filename for Google Trends file

heatmap_argo *Heatmap plot of ARGO coefficients applied on CDC's ILI data*

Description

Heatmap plot of ARGO coefficients applied on CDC's ILI data

Usage

```
heatmap_argo(argo_coef, lim = 0.1, na.grey = TRUE, scale = 1)
```

Arguments

argo_coef	The coefficient matrix
lim	the limit to truncate for large coefficients for better presentation
na.grey	whether to plot grey for NA values
scale	margin scale

Value

a graph on the default plot window

Examples

```
cor_coef <- matrix(runif(100, -1, 1), ncol=10)
colnames(cor_coef) <- as.character(Sys.Date() - 10:1)
rownames(cor_coef) <- paste0("row", 1:10)
pdf(file.path(tempdir(), "heatmap_argo.pdf"), height=11,width=12)
heatmap_argo(cor_coef)
dev.off()
```

`heatmap_cor` *Heatmap plot of correlation matrix*

Description

Heatmap plot of correlation matrix

Usage

```
heatmap_cor(cor_heat, lim = 1)
```

Arguments

`cor_heat` The coefficient matrix to draw heatmap

`lim` the limit to truncate for large coefficients for better presentation

Value

a graph on the default plot window

Examples

```
cor_coef <- matrix(runif(100, -1, 1), ncol=10)
colnames(cor_coef) <- paste0("col", 1:10)
rownames(cor_coef) <- paste0("row", 1:10)
heatmap_cor(cor_coef)
```

`load_data` *Parsing of raw data*

Description

Data related to the PNAS paper. Accessed on Nov 14, 2015.

Usage

```
load_data(type = "extdata", ili.weighted = TRUE)
```

Arguments

`type` the type of the data to be loaded. If `type=="extdata"` it loads the data to reproduce the PNAS paper, and if `type=="athdata"` it loads the data to reproduce the CID(?) paper.

`ili.weighted` logical indicator to specify whether to load weighted ILI or not, if `FALSE` unweighted ILI is loaded.

Details

Parse and load CDC's ILI data, Google Flu Trend data, Google Correlate data trained with ILI as of 2010, Google Correlate data trained with ILI as of 2009, Google Trend data with search terms identified from Google Correlate (2010 version).

Each week ends on the Saturday indicated in the xts object

Google Correlate data is standardized by Google, and we rescale it to 0 – 100 during parsing. Google Trends data is in the scale of 0 – 100.

Value

A list of following named xts objects if `type=="extdata"`

- `GC10` Google Correlate trained with ILI available as of 2010. Google Correlate has been deprecated by Google as of Dec 2019 and is no longer publicly available.
- `GC09` Google Correlate trained with ILI available as of 2009.
- `GT` Google Trends data for search queries identified using Google Correlate. Not directly available online, you have to manually input query terms at https://trends.google.com/trends/
- `CDC` CDC's ILI dataset. Available online at https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
- `GFT` Google Flu Trend (historical predictions).

A list of following named xts objects if `type=="athdata"`

- `GT` Google Trends data for search queries identified. Not directly available online, you have to manually input query terms at https://trends.google.com/trends/
- `CDC` CDC's ILI dataset. Available online at https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
- `ili_idx` the indexing information that includes the week number and year number, the date of ending Saturday, and the season number Available online at https://www.cdc.gov/flu/weekly/
- `ATH` Athenahealth data that includes the proportion of "Flu Visit", "ILI Visit", and "Unspecified Viral or ILI Visit" compared to total number of visit to the Athenahealth partner healthcare providers.
- `ili_unrevised` Historical unrevised ILI activity level. The unrevised ILI published on week ZZ of season XXXX-YYYY is available at `www.cdc.gov/flu/weekly/weeklyarchivesXXXX-YYYY/data/senAllreg` or `.htm`. For example, original ILI report for week 7 of season 2015-2016 is available at https://www.cdc.gov/flu/weekly/weeklyarchives2015-2016/data/senAllregt07.html, and original ILI report for week 50 of season 2012-2013 is available at https://www.cdc.gov/flu/weekly/weeklyarchives2012-2013/data/senAllregt50.htm

References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

Examples

```
system.file("extdata", "correlate-Influenza_like_Illness_h1n1_CDC_.csv", package = "argo")
system.file("extdata", "correlate-Influenza_like_Illness_CDC_.csv", package = "argo")
system.file("extdata", "GFT.csv", package = "argo")
system.file("extdata", "ILINet.csv", package = "argo")
load_data()
```

`load_reg_data` *Parsing of raw data for regional ILI estimation*

Description

Parsing of raw data for regional ILI estimation

Usage

```
load_reg_data(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  gt.parser = gt.parser.pub.web
)
```

Arguments

`gt.folder` folder with all Google Trends data

`ili.folder` folder with all ILI data

`population.file` csv file path with state population data

`gft.file` csv file path for Google Flu Trends

`gt.parser` Google Trends data parser function, could be ‘gt.parser.pub.web‘ or ‘gt.parser.pub.api‘

References

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

Examples

```
download.file("https://scholar.harvard.edu/files/syang/files/gt2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")
```

```
data_parsed <- load_reg_data(
  gt.folder=gt.folder,
  ili.folder=system.file("regiondata", "ili20161121", package = "argo"),
  population.file=system.file("regiondata", "Population.csv", package = "argo"),
  gft.file=system.file("regiondata", "GFT.txt", package = "argo")
)
```

logit *logit function*

Description

logit function

Usage

```
logit(x)
```

Arguments

x	numeric value for logit transformation

Examples

```
logit(0.5)
```

logit_inv *inverse logit function*

Description

inverse logit function

Usage

```
logit_inv(x)
```

Arguments

x	numeric value for inverse logit transformation

Examples

```
logit_inv(0)
```

parse_gt_weekly *Parsing of Google Trends data*

Description

Parsing of Google Trends data

Usage

```
parse_gt_weekly(folder)
```

Arguments

folder	folder with weekly Google Trends file

References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

Examples

```


download.file("https://scholar.harvard.edu/files/syang/files/gt2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")
parsed_data <- parse_gt_weekly(gt.folder)
```

parse_unrevised_ili *Parsing of unrevised ili from online source*

Description

Parsing of unrevised ili from online source

Usage

```
parse_unrevised_ili(type = "extdata", ili.weighted = TRUE)
```

Arguments

type	the type of data folder to parse
ili.weighted	indicator to use weighted ILI or not

References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

Examples

```
parse_unrevised_ili()
```

plot_argo *Time series plot of ARGO applied on CDC's ILI data*

Description

This function is used to reproduce the ARGO plot.

Usage

```
plot_argo(GFT_xts, GC_GT_cut_date, model_names, legend_names, zoom_periods)
```

Arguments

GFT_xts	dataframe with all predicted values
GC_GT_cut_date	cutting date for switching datasets
model_names	name of predicting models
legend_names	legend for predicting models
zoom_periods	vector of periods to zoom into

Value

a graph on the default plot window

Examples

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=5)), order.by = Sys.Date() - (200:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"
zoom_periods = c()
for (i in 0:5){
  zoom_periods = c(
    zoom_periods,
    paste0(zoo::index(GFT_xts)[i*30+1], "/", zoo::index(GFT_xts)[i*30+30])
  )
}
```

```
plot_argo(
  GFT_xts = GFT_xts,
  GC_GT_cut_date = zoo::index(GFT_xts)[50],
  model_names = colnames(GFT_xts)[-1],
  legend_names = paste0(colnames(GFT_xts)[-1], "legend"),
  zoom_periods = zoom_periods
)
```

`summary_argo` *performance summary of ARGO applied on CDC's ILI data*

Description

performance summary of ARGO applied on CDC's ILI data

Usage

```
summary_argo(
  GFT_xts,
  model_names,
  legend_names,
  periods,
  whole_period = "2009-03/2015-10"
)
```

Arguments

`GFT_xts`	dataframe with all predicted values
`model_names`	name of predicting models
`legend_names`	legend for predicting models
`periods`	vector of periods to zoom into
`whole_period`	the whole period duration

Value

A list of summary tables for the input periods, including RMSE, MAE, MAPE, corr

References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.
Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

Examples

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=10)), order.by = Sys.Date() - (100:1))
names(GFT_xts) <- paste0("col", 1:10)
names(GFT_xts)[1] <- "CDC.data"
summary_argo(
  GFT_xts = GFT_xts,
  model_names = colnames(GFT_xts)[-1],
  legend_names = paste0(colnames(GFT_xts)[-1], "legend"),
  periods = c(paste0(zoo::index(GFT_xts)[1], "/", zoo::index(GFT_xts)[49]),
              paste0(zoo::index(GFT_xts)[50], "/", zoo::index(GFT_xts)[100])),
  whole_period="2009-03/"
)
```

Index