Package ‘GPP’

November 23, 2020

Title Estimating a Counterfactual with Uncertainty Using Gaussian Process Projection

Version 0.1

Description Estimates a counterfactual using Gaussian process projection. It takes a dataframe, creates missingness in the desired outcome variable and estimates counterfactual values based on all information in the dataframe. The package writes Stan code, checks it for convergence and adds artificial noise to prevent overfitting and returns a plot of actual values and estimated counterfactual values using r-base plot.

Depends R (>= 3.5.0), methods, rstan, parallel

LazyData true

Encoding UTF-8

License GPL (>= 2)

RoxygenNote 7.1.1

NeedsCompilation no

Author Devin P. Brown [aut],
David Carlson [aut, cre]

Maintainer David Carlson <carlson.david@wustl.edu>

R topics documented:

autoConverge *Checks Stan model for convergence, then runs model on actual data.*

Description

Return a converged Stan model fit and the recommended noise level.

Usage

```
autoConverge(
  df,
  controlVars,
  nUntreated,
  obvColName,
  obvName,
  outcomeName,
  starttime,
  timeColName,
  filepath = NULL,
  ncores = NULL,
  iter = 25000,
  epsilon = 0.02,
  noise = 0.1,
  printMod = FALSE,
  shift = 0.05
)
```

Arguments

`df` The dataframe used for the model.

`controlVars` String of column names for control variables.

`nUntreated` The number of untreated units in the model.

`obvColName` The column name that includes the observation subject to the counterfactual.

`obvName` The name of the observation subject to the counterfactual.

`outcomeName` The outcome variable of interest.

`starttime` The start time of the counterfactual estimation.

`timeColName` The name of the column that includes the time variable.

`filepath` Your preferred place to save the fit data. See Details.

`ncores` The number of cores to be used to run the model. Default of NULL will utilize all cores.

`iter` Preferred number of iterations. See details.

`epsilon` The desired level of convergence, i.e. how close to the 0.95 coverage is acceptable.

`noise` The baseline level of noise to be added to the model to prevent overfit. Updates as the model runs.

`printMod` Boolean. Defaults FALSE. If TRUE, prints the model block for the run to the console. See details.

`shift` The magnitude of adjustment for the noise level per iteration. Defaults to 0.05.

Details

We recommend creating a new folder for the file path since the Stan fit creates a large number of files at runtime.

For iterations, check that your model converged (we recommend all r-hats close to 1 and examining traceplots).

We recommend keeping printMod as FALSE, otherwise, the function will write the model to the console for every model run on the convergence.

We also recommend using all cores on your machine to speed up model run time. If you are unsure about the number of cores in your machine, see doParallel::detectCores().

Value

The recommended noise level after convergence.

Author(s)

Devin P. Brown <`devinpbrown96@gmail.com`> and David Carlson <`carlson.david@wustl.edu`>

See Also

`plotGPPfit` `runMod` `GPP` `writeMod`

`GDPdata` *1960-2003 GDP dataset*

Description

An example dataset for using `GPP` to estimate the counterfactual GDP of West Germany assuming no reunification.

Usage

```
GDPdata
```

Format

A data frame with 748 rows and 14 columns. For detailed explanations of the exact measures, see https://www.dropbox.com/s/n1bvqb54xrw8vyj/GPSynth.pdf?dl=0:

index

country

year

gdp

infrate

trade

schooling

invest60

invest70

invest80

industry

invest

school

ind

See Also

GPP plotGPPfit writeMod runMod autoConverge

GPP *Estimates a counterfactual with uncertainty using Gaussian process projection*

Description

Returns a list of a plot object (after making the plot) of estimated counterfactual values after checking for model convergence and adjusting the noise level, and returns the fitted model.

Usage

```
GPP(
  df,
  controlVars,
  nUntreated,
  obvColName,
  obvName,
  outcomeName,
  starttime,
  timeColName,
  ncores = NULL,
  epsilon = 0.02,
  noise = 0.1,
  printMod = FALSE,
  shift = 0.05,
  iter = 25000,
  filepath = NULL,
  legendLoc = ”topleft”,
  xlabel = NULL,
  ylabel = NULL,
  actualdatacol = ”black”,
  preddatacol = ”red”,
  ...
)
```

Arguments

`df` The dataframe used for the model.

`controlVars` String of column names for control variables.

`nUntreated` The number of untreated units in the model.

`obvColName` The column name that includes the observation subject to the counterfactual.

`obvName` The name of the observation subject to the counterfactual.

`outcomeName` The outcome variable of interest.

`starttime` The start year of the counterfactual estimation.

`timeColName` The name of the column that includes the time variable.

`ncores` The number of cores to be used to run the model. See details.

`epsilon` The desired level of convergence.

`noise` The baseline level of noise to be added to the model to prevent overfit. Updates as the model runs.

`printMod` Boolean. Defaults FALSE. If TRUE, prints each model block to the console. See details.

`shift` The magnitude of adjustment for the noise level per iteration. Defaults to 0.05.

`iter` The number of iterations you would like to run. Defaults to 25,000. See details.

`filepath` Your preferred place to save the fit data. See Details.

`legendLoc` The preferred location of the legend in the final graph. Defaults to "topleft".

`xlabel` The label of the x-axis in the final graph. Defaults to input for ’timeColName’.

`ylabel` The preferred label of the y-axis in the final graph. Defaults to input for ’outcomeName’.

`actualdatacol` The preferred color for plotted line for actual data. Defaults to black.

`preddatacol` The preferred color for plotted line for predicted counterfactual data. Defaults to red.

`...` Further parameters passed to the plot function.

Details

We recommend using all cores on your machine to speed up model run time. If you are unsure about the number of cores in your machine, see `parallel::detectCores()`.

We recommend keeping printMod as FALSE, otherwise, the function will write the model to the console for every model run on the convergence.

For iterations, check that your model converged (we recommend all r-hats close to 1 and examining traceplots).

We recommend creating a new folder for the file path since the Stan fit creates a large number of files at runtime.

Value

A plot of the actual values and the estimated counterfactual values of the model, and the final model fit.

Author(s)

Devin P. Brown <devinpbrown96@gmail.com> and David Carlson <carlson.david@wustl.edu>

See Also

plotGPPfit writeMod runMod autoConverge

Examples

```
## Not run:
data(GDPdata)
out = GPP(df = GDPdata,
    controlVars = c('invest', 'school', 'ind'),
    nUntreated = length(unique(GDPdata$country))-1,
    obvColName = 'country', obvName = 'West Germany',
    outcomeName = 'gdp', starttime = 1989,
    timeColName = 'year')

## End(Not run)
```

plotGPPfit *Plots results of a (converged) model, with true and projected values.*

Description

Takes the results of a Gaussian Process Projection fit and generates a linear plot of the actual and predicted counterfactual values

Usage

```
plotGPPfit(
  fit,
  df,
  obvColName,
  obvName,
  outcomeName,
  starttime,
  timeColName,
  legendLoc = "topleft",
  xlabel = NULL,
  ylabel = NULL,
  actualdatacol = "black",
  preddatacol = "red",
  ...
)
```

Arguments

`fit` The fit results of the GPP stan model.

`df` The dataframe used in your model.

`obvColName` The column name that includes your observation of interest. Must be a string.

`obvName` The name of the specific observation of interest. Must be a string.

`outcomeName` The explanatory variable that is subjected to the counterfactual claim.

`starttime` The start time of the treatment effect.

`timeColName` The name of the column that includes your time variable.

`legendLoc` The preferred location of the legend in the final graph. Defaults to "topleft".

`xlabel` The label of the x-axis in the final graph. Defaults to input for 'timeColName'.

`ylabel` The preferred label of the y-axis in the final graph. Defaults to input for 'outcomeName'.

`actualdatacol` The preferred color for plotted line for actual data. Defaults to black.

`preddatacol` The preferred color for plotted line for predicted counterfactual data. Defaults to red.

`...` Further graphical parameters.

Value

A plot built in r-base

Author(s)

Devin P. Brown <`devinpbrown96@gmail.com`> and David Carlson <`carlson.david@wustl.edu`>

See Also

`autoConverge` `GPP` `runMod` `writeMod`

Examples

```
## Not run:
load(FDIout)
load(d)
df = d
rm(list=d)

plotGPPfit(fit=fit, df=df, obvColName='country',
    obvName='West Germany',outcomeName='gdp', starttime=1990,
    timeColName='year', legendLoc='bottomright',
    xlabel=”Test X-Label”, ylabel = ”Test Y-Label”,
    actualdatacol = 'blue', preddatacol = 'green')

## End(Not run)
```

---

runMod *Runs the model, given the data and treated case (may be a placebo).*

Description

Returns a fit of the Stan model for all observations.

Usage

```
runMod(modText, dataBloc, unit, iter = 25000, filepath = NULL)
```

Arguments

`modText` This is the string that contains your Stan code. Can be written with `writeMod`.

`dataBloc` This is the data that you pass to the Stan code. It is automatically generated when you run `autoConverge`.

`unit` The unit of observation to project.

`iter` The number of iterations you would like to run. Defaults to 25,000.

`filepath` Your preferred place to save the fit data. See Details.

Details

For iterations, check that your model converged (we recommend all r-hats close to 1 and examining traceplots).

We recommend creating a new folder for the file path since the Stan fit creates a large number of files at runtime.

Value

The fit for the GPP counterfactual Stan model.

Author(s)

Devin P. Brown <devinpbrown96@gmail.com> and David Carlson <carlson.david@wustl.edu>

See Also

plotGPPfit writeMod GPP autoConverge

---

writeMod *Writes Stan code for GPP model*

Description

Returns string of Stan code that can be run to estimate the GPP.

Usage

```
writeMod(noise, ncov, printMod = FALSE)
```

Arguments

noise	The desired amount of artificial noise to add to the model.
ncov	The number of covariates to include in the model.
printMod	Boolean. Defaults FALSE. If TRUE, prints each model block to the console. See details.

Details

We recommend keeping printMod as FALSE, otherwise, the function will write the model to the console for every model run on the convergence.

Value

A string of Stan code that can be run with runMod

Author(s)

Devin P. Brown <devinpbrown96@gmail.com> and David Carlson <carlson.david@wustl.edu>

See Also

plotGPPfit runMod GPP autoConverge

Examples

```
## Not run:
writeMod(noise = 0.25, ncov = 2)

## End(Not run)
```

Index