Package ‘hdpGLM’

May 7, 2022

Title  Hierarchical Dirichlet Process Generalized Linear Models
Version  1.0.2
Depends  R (>= 3.3.3)
License  MIT + file LICENSE
URL  https://github.com/DiogoFerrari/hdpGLM,
     http://www.diogoferrari.com/hdpGLM/index.html
BugReports  https://github.com/DiogoFerrari/hdpGLM/issues
Encoding  UTF-8
LazyData  true
LinkingTo  Rcpp, RcppArmadillo
Imports  coda, data.table, dplyr, formula.tools, ggjoy, ggplot2, stringr, ggridges, ggpublisher, Hmisc, isotone, questionr, LaplacesDemon, magrittr, methods, MASS, MCMCpack, mvtnorm, Rcpp, rprojroot, purrr, tibble, tidyr, tidyverse
RoxygenNote  7.1.2
Suggests  knitr, rmarkdown
VignetteBuilder  knitr
NeedsCompilation  yes
Author  Diogo Ferrari [aut, cre]
Maintainer  Diogo Ferrari <diogoferrari@gmail.com>
Repository  CRAN
Date/Publication  2022-05-07 00:10:02 UTC
R topics documented:

classify ................................................................. 2  
coef.dpGLM .............................................................. 3  
coef.hdpGLM ............................................................. 3  
hdpGLM ................................................................. 4  
hdpGLM_classify ........................................................ 6  
hdpGLM_package ......................................................... 7  
hdpGLM_simParameters .................................................. 8  
hdpGLM_simulateData .................................................... 9  
mcmc_info.dpGLM ........................................................ 11  
mcmc_info.hdpGLM ....................................................... 11  
nclusters ............................................................... 11  
plot.dpGLM .............................................................. 12  
plot.hdpGLM ............................................................. 14  
plot_beta ............................................................... 16  
plot_beta_sim ........................................................... 17  
plot_hdpGlm ............................................................ 18  
plot_pexp_beta .......................................................... 20  
plot_tau ................................................................. 23  
predict.dpGLM ........................................................... 25  
predict.hdpGLM .......................................................... 25  
print.dpGLM ............................................................. 26  
print.dpGLM_data ....................................................... 27  
print.hdpGLM ............................................................ 27  
print.hdpGLM_data ..................................................... 28  
summary.dpGLM .......................................................... 28  
summary.dpGLM_data ..................................................... 29  
summary.hdpGLM ........................................................ 29  
summary.hdpGLM_data ................................................... 30  
summary_tidy ............................................................ 31  
welfare ................................................................. 31  
welfare2 ................................................................. 32  

Index 33

classify Classify data points

Description

This function returns a data frame with the data points classified according to the estimation of cluster probabilities generated by the output of the function hdpGLM.

Usage

classify(data, samples)
coef.dpGLM

Arguments

- data: a data frame with the data set used to estimate the hdpGLM model
- samples: the output of hdpGLM

Description

This function gives the posterior mean of the coefficients

Usage

```r
## S3 method for class 'dpGLM'
coef(object, ...)
```

Arguments

- object: a dpGLM object returned by the function hdpGLM
- ...: The additional parameters accepted are:

coef.hdpGLM

Extract dpGLM fitted coefficients

Description

This function gives the posterior mean of the coefficients

Usage

```r
## S3 method for class 'hdpGLM'
coef(object, ...)
```

Arguments

- object: a dpGLM object returned by the function hdpGLM
- ...: The additional parameters accepted are:
hdpGLM

Fit Hierarchical Dirichlet Process GLM

Description

The function estimates a semi-parametric mixture of Generalized Linear Models. It uses a (hierarchical) Dependent Dirichlet Process Prior for the mixture probabilities.

Usage

hdpGLM(
  formula1,
  formula2 = NULL,
  data,
  mcmc,
  family = "gaussian",
  K = 100,
  context.id = NULL,
  constants = NULL,
  weights = NULL,
  n.display = 1000,
  na.action = "exclude",
  imp.bin = "R"
)

Arguments

formula1
  a single symbolic description of the linear model of the mixture GLM components to be fitted. The syntax is the same as used in the \texttt{lm} function.

formula2
  either NULL (default) or a single symbolic description of the linear model of the hierarchical component of the model. It specifies how the average parameter of the base measure of the Dirichlet Process Prior varies linearly as a function of group level covariates. If NULL, it will use a single base measure to the DPP mixture model.

data
  a data.frame with all the variables specified in formula1 and formula2. Note: it is advisable to scale the variables before the estimation

mcmc
  a named list with the following elements
    - burn.in (required): an integer greater or equal to 0 indicating the number of iterations used in the burn-in period of the MCMC.
    - n.iter (required): an integer greater or equal to 1 indicating the number of iterations to record after the burn-in period for the MCMC.
    - epsilon (optional): a positive number. Default is 0.01. Used when family='binomial' or family='multinomial'. It is used in the Stormer-Verlet Integrator (a.k.a leapfrog integrator) to solve the Hamiltonian Monte Carlo in the estimation of the model.
- `leapFrog` (optional) an integer. Default is 40. Used when `family`='binomial' or `family`='multinomial'. It indicates the number of steps taken at each iteration of the Hamiltonian Monte Carlo for the Stormer-Verlet Integrator.

- `hmc_iter` (optional) an integer. Default is 1. Used when `family`='binomial' or `family`='multinomial'. It indicates the number of HMC iteration(s) for each Gibbs iteration.

`family` a character with either 'gaussian', 'binomial', or 'multinomial'. It indicates the family of the GLM components of the mixture model.

`K` an integer indicating the maximum number of clusters to truncate the Dirichlet Process Prior in order to use the blocked Gibbs sampler.

`context.id` string with the name of the column in the data that uniquely identifies the contexts. If `NULL` (default) contexts will be identified by numerical indexes and unique context-level variables. The user is advised to pre-process the data to provide meaningful labels for the contexts to facilitate later visualization and analysis of the results.

`constants` either `NULL` or a list with the constants of the model. If not `NULL`, it must contain a vector named `mu_beta`, whose size must be equal to the number of covariates specified in `formula1` plus one for the constant term; `Sigma_beta`, which must be a squared matrix, and each dimension must be equal to the size of the vector `mu_beta`; and `alpha`, which must be a single number. If `@param family` is 'gaussian', then it must also contain `s2_sigma` and `df_sigma`, both single numbers. If `NULL`, the defaults are `mu_beta=0, Sigma_beta=diag(10), alpha=1, df_sigma=10, s2_sigma=10` (all with the dimension automatically set to the correct values).

`weights` numeric vector with the same size as the number of rows of the data. It must contain the weights of the observations in the data set. NOTE: FEATURE NOT IMPLEMENTED YET

`n.display` an integer indicating the number of iterations to wait before printing information about the estimation process. If zero, it does not display any information. Note: displaying information at every iteration (`n.display=1`) may increase the time to estimate the model slightly.

`na.action` string with action to be taken for the NA values. (currently, only `exclude` is available)

`imp.bin` string, either "R" or "Cpp" indicating the language of the implementation of the binomial model.

**Details**

This function estimates a Hierarchical Dirichlet Process generalized linear model, which is a semi-parametric Bayesian approach to regression estimation with clustering. The estimation is conducted using Blocked Gibbs Sampler if the output variable is gaussian distributed. It uses Metropolis-Hastings inside Gibbs if the output variable is binomial or multinomial distributed. This is specified using the parameter `family`. See:


Value

The function returns a list with elements samples, pik, max_active, n.iter, burn.in, and time.elapsed. The samples element contains a MCMC object (from coda package) with the samples from the posterior distribution. The pik is a n x K matrix with the estimated probabilities that the observation $i$ belongs to the cluster $k$.

Examples

```r
## Note: this example is for illustration. You can run the example
## manually with increased number of iterations to see the actual
## results, as well as the data size (n)
set.seed(10)
n = 300
data = tibble::tibble(x1 = rnorm(n, -3),
                      x2 = rnorm(n, 3),
                      z = sample(1:3, n, replace=TRUE),
                      y =I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                      I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                      I(z==3) * (3 - 4*x1 - x2 + rnorm(n))
                        )

mcmc = list(burn.in = 0, n.iter = 20)
samples = hdpGLM(y~ x1 + x2, data=data, mcmc=mcmc, family='gaussian',
                 n.display=30, K=50)
summary(samples)
plot(samples)
plot(samples, separate=TRUE)

## compare with GLM
## lm(y~ x1 + x2, data=data, family='gaussian')
```

### Description

Deprecated
Usage

hdpGLM_classify(data, samples)

Arguments

data    a data frame with the data set used to estimate the hdpGLM model
samples the output of hdpGLM

Description

Further information is available at: http://www.diogoferrari.com/hdpGLM/index.html

References:

Details

The package implements a hierarchical Dirichlet process Generalized Linear Model as proposed in Ferrari (2020) Modeling Context-Dependent Latent Effect Heterogeneity, which expands the non-parametric Bayesian models proposed in Mukhopadhyay and Gelfand (1997), Hannah (2011), and Heckman and Vytlacil (2007) to deal with context-dependent cases. The package can be used to estimate latent heterogeneity in the marginal effect of GLM linear coefficients, to cluster data points based on that latent heterogeneity, and to investigate the occurrence of Simpson’s Paradox due to latent or omitted features.
Simulate the parameters of the model

**Description**

This function generates parameters that can be used to simulate data sets from the Hierarchical Dirichlet Process of Generalized Linear Model (hdpGLM) or dpGLM.

**Usage**

```r
hdpGLM_simParameters(
  K,
  nCov = 2,
  nCovj = 0,
  J = 1,
  pi = NULL,
  same.K = FALSE,
  seed = NULL,
  context.effect = NULL,
  same.clusters.acrosscontexts = NULL,
  context.dependent.cluster = NULL
)
```

**Arguments**

- `K`: integer, the number of clusters. If there are multiple contexts, `K` is the average number of clusters across contexts, and each context gets a number of clusters sampled from a Poisson distribution, except if `same.K` is `TRUE`.
- `nCov`: integer, the number of covariates of the GLM components
- `nCovj`: an integer indicating the number of covariates determining the average parameter of the base measure of the Dirichlet process prior
- `J`: an integer representing the number of contexts @param parameters either `NULL` or a list with the parameters to generate the model. If not `NULL`, it must contain a sublist name `beta`, a vector named `tau`, and a vector named `pi`. The sublist `beta` must be a list of vectors, each one with size `nCov+1` to be the coefficients of the GLM mixtures components that will generate the data. For the vector `tau`, if `nCovj=0` (single-context case) then it must be a 1x1 matrix containing 1. If `nCovj>0`, it must be a `(nCov+1)x(nCovj+1)` matrix. The vector `pi` must add up to 1 and have length `K`.
- `pi`: either `NULL` or a vector with length `K` that add up to 1. If not `NULL`, it determines the mixture probabilities
- `same.K`: boolean, used when data is sampled from more than one context. If `TRUE` all contexts get the same number of clusters. If `FALSE`, each context gets a number of clusters sampled from a Poisson distribution with expectation equals to `K` (current not implemented)
seed
a seed for set.seed
context.effect
either NULL or a two dimensional integer vector. If it is NULL, all the coefficients (beta) of the individual level covariates are functions of context-level features (tau). If it is not NULL, the first component of the vector indicates the index of the lower level covariate (X) whose linear effect beta depends on context (tau) (0 is the intercept). The second component indicates the index context-level covariate (W) whose linear coefficient (tau) is non-zero.
same.clusters.across.contexts
boolean, if TRUE all the contexts will have the same number of clusters AND each cluster will have the same coefficient beta.
context.dependent.cluster
integer, indicates which cluster will be context-dependent. If zero, all clusters will be context-dependent

Value
The function returns a list with the parameters used to generate data sets from the hdpGLM model. This list can be used in the function hdpGLM_simulateData

Examples
pars = hdpGLM_simParameters(nCov=2, K=2, nCovj=3, J=20, 
same.clusters.across.contexts=FALSE, context.dependent.cluster=0)

hdpGLM_simulateData
Simulate a Data Set from hdpGLM

Description
Simulate a Data Set from hdpGLM

Usage
hdpGLM_simulateData(
  n,
  K,
  nCov = 2,
  nCovj = 0,
  J = 1,
  family = "gaussian",
  parameters = NULL,
  pi = NULL,
  same.K = FALSE,
  seed = NULL,
  context.effect = NULL,
  same.clusters.across.contexts = NULL,
  context.dependent.cluster = NULL
)
**Arguments**

- **n** integer, the sample size of the data. If there are multiple contexts, each context will have n cases.

- **K** integer, the number of clusters. If there are multiple contexts, K is the average number of clusters across contexts, and each context gets a number of clusters sampled from a Poisson distribution, except if same.K is TRUE.

- **nCov** integer, the number of covariates of the GLM components.

- **nCovj** an integer indicating the number of covariates determining the average parameter of the base measure of the Dirichlet process prior

- **J** an integer representing the number of contexts @param parameters either NULL or a list with the parameters to generate the model. If not NULL, it must contain a sublist name beta, a vector named tau, and a vector named pi. The sublist beta must be a list of vectors, each one with size nCov+1 to be the coefficients of the GLM mixtures components that will generate the data. For the vector tau, if nCovj=0 (single-context case) then it must be a 1x1 matrix containing 1. If nCovj>0, it must be a (nCov+1)x(nCovj+1) matrix. The vector pi must add up to 1 and have length K. 

- **family** a character with either 'gaussian', 'binomial', or 'multinomial'. It indicates the family of the GLM components of the mixture model.

- **parameters** a list with the parameter values of the model. Format should be the same of the output of the function hdpGLM_simulateParameters()

- **pi** either NULL or a vector with length K that add up to 1. If not NULL, it determines the mixture probabilities

- **same.K** boolean, used when data is sampled from more than one context. If TRUE all contexts get the same number of clusters. If FALSE, each context gets a number of clusters sampled from a Poisson distribution with expectation equals to K (current not implemented)

- **seed** a seed for set.seed

- **context.effect** either NULL or a two dimensional integer vector. If it is NULL, all the coefficients (beta) of the individual level covariates are functions of context-level features (tau). If it is not NULL, the first component of the vector indicates the index of the lower level covariate (X) whose linear effect beta depends on context (tau) (0 is the intercept). The second component indicates the index context-level covariate (W) whose linear coefficient (tau) is non-zero.

- **same.clusters.across.contexts** boolean, if TRUE all the contexts will have the same number of clusters AND each cluster will have the same coefficient beta.

- **context.dependent.cluster** integer, indicates which cluster will be context-dependent. If zero, all clusters will be context-dependent
mcmc_info.dpGLM

Description
Generic method to return the MCMC information

Usage
mcmc_info.dpGLM(x, ...)

Arguments
x  a dpGLM object returned by the function hdpGLM
... ignore

mcmc_info.hdpGLM

Description
Generic method to return the MCMC information

Usage
mcmc_info.hdpGLM(x, ...)

Arguments
x  a hdpGLM object returned by the function hdpGLM
... ignore

nclusters

Description
This function returns the number of clusters found in the estimation

Usage
nclusters(object)

Arguments
object  a dpGLM object returned by the function hdpGLM
plot.dpGLM

Default plot for class dpGLM

Description
This function generates density plots with the posterior distribution generated by the function hdpGLM.

Usage
## S3 method for class 'dpGLM'
plot(
x, 
terms = NULL, 
separate = FALSE, 
hpd = TRUE, 
true.beta = NULL, 
title = NULL, 
subtitle = NULL, 
adjust = 1, 
ncols = NULL, 
only.occupied.clusters = TRUE, 
focus.hpd = FALSE, 
legend.position = "top", 
colour = "grey", 
alpha = 0.4, 
display.terms = TRUE, 
plot.mean = TRUE, 
legend.label.true.value = "True", 
...
)

Arguments

x a dpGLM object with the samples from generated by hdpGLM

terms string vector with the name of covariates to plot. If NULL (default), all covariates are plotted.

separate boolean, if TRUE the linear coefficients beta will be displayed in their separate clusters.

hpd boolean, if TRUE and separate=T, the 95% HPDI lines will be displayed.

true.beta either NULL (default) or a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named k indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx, where dx is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named True with the true value of the betas.
plot.dpGLM

- **title**: string, the title of the plot
- **subtitle**: string, the subtitle of the plot
- **adjust**: the bandwidth used is actually adjust*bw. This makes it easy to specify values like ‘half the default’ bandwidth.
- **ncols**: integer, the number of columns in the plot
- **only.occupied.clusters**: boolean, if TRUE it shows only the densities of the clusters that actually have data points assigned to it with high probability
- **focus.hpd**: boolean, if TRUE and separate is also TRUE it will display only the 95% HPDI of the posterior density of the linear coefficients beta
- **legend.position**: one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend
- **colour**: string with color to fill the density plot
- **alpha**: number between 0 and 1 indicating the degree of transparency of the density display.
- **display.terms**: boolean, if TRUE (default), the covariate name is displayed in the plot
- **plot.mean**: boolean, if TRUE the posterior mean of every cluster is displayed
- **legend.label.true.value**: a string with the value to display in the legend when the true.beta is used

... ignored

**Examples**

```r
# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data = tibble::tibble(x1 = rnorm(n, -3),
x2 = rnorm(n, 3),
z = sample(1:3, n, replace=TRUE),
y =I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
I(z==3) * (3 - 4*x1 - x2 + rnorm(n)) ,
}

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, data=data, mcmc=mcmc, n.display=1)
plot(samples)
```
Description

Generic function to plot the posterior density estimation produced by the function hdpGLM

Usage

```r
## S3 method for class 'hdpGLM'
plot(
x,
terms = NULL,
j.label = NULL,
j.idx = NULL,
title = NULL,
subtitle = NULL,
true.beta = NULL,
col = NULL,
legend.position = "bottom",
display.terms = TRUE,
context.id = NULL,
ylab = NULL,
xlab = NULL,
x.axis.size = 1.1,
y.axis.size = 1.1,
title.size = 1.1,
panel.title.size = 1.5,
legend.size = 1.1,
rel.height = 0.01,
fill.col = "#00000044",
border.col = "white",
...
)
```

Arguments

- **x**: an object of the class hdpGLM generated by the function hdpGLM
- **terms**: string vector with the name of the individual-level covariates to plot. If NULL (default), all covariates are plotted.
- **j.label**: string vector with the names of the contexts to plot. An alternative is to use the context indexes with the parameter j.idx instead of the context labels. If NULL (default) and j.idx is also NULL, the posterior distribution of all contexts are plotted. Note: if contexts to plot are selected using j.label, the parameter context.id must also be provided.
j.idx: integer vector with the index of the contexts to plot. An alternative is to use the context labels with the parameter j.label instead of the indexes. If NULL (default) and j.label is also NULL, the posterior distribution of all contexts are plotted.

title: string, the title of the plot

subtitle: string, the subtitle of the plot

true.beta: a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named j with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named k must be provided, indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx, where dx is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named True with the true value of the betas. Finally, the data.frame must contain columns with the context-level covariates as used in the estimation of the hdpGLM function (see Details below).

ncol: integer, the number of columns in the plot

legend.position: one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend

display.terms: boolean, if TRUE (default), the covariate name is displayed in the plot

context.id: string with the name of the column containing the labels identifying the contexts. This variable should have been specified when the estimation was conducted using the function hdpGLM.

ylab: string, the label of the y-axis

xlab: string, the label of the x-axis

x.axis.size: numeric, the relative size of the label in the x-axis

y.axis.size: numeric, the relative size of the label in the y-axis

title.size: numeric, the relative size of the title of the plot

panel.title.size: numeric, the relative size of the titles in the panel of the plot

legend.size: numeric, the relative size of the legend

rel.height: see ggridges::geom_density_ridges

fill.col: string with the color of the densities

border.col: string with the color of the border of the densities

... Additional arguments accepted are:

true.beta: a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named j with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named k must be provided, indicating the cluster of beta, and a column named Parameter with the name of the linear coefficients (beta1, beta2, ..., beta_dx, ...,
where dx is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named True with the true value of the betas. Finally, the data.frame must contain columns with the context-level covariates as used in the estimation of the hdpGLM function (see Details below).

ture.tau: a data.frame with four columns. The first must be named w and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named beta must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named Parameter must be named tau<sub>w</sub><sub>&beta;</sub>, where w and beta must be the actual values displayed in the columns w and beta. Finally, it must have a column named True with the true value of the parameter.

---

**plot_beta**

*Plot beta posterior distribution*

**Description**

Plot the posterior distribution of the linear parameters beta for each context

**Usage**

```r
plot_beta(
samples,
X = NULL,
context.id = NULL,
true.beta = NULL,
title = NULL,
subtitle = NULL,
plot.mean = FALSE,
plot.grid = FALSE,
showKhat = FALSE,
col = NULL,
xlab.size = NULL,
ylab.size = NULL,
title.size = NULL,
legend.size = NULL,
xtick.distance = NULL,
left.margin = 0,
ytick.distance = NULL,
col.border = "white"
)
```

**Arguments**

- `samples` an output of the function hdpGLM
plot_beta_sim

X
context.id
true.beta
title
subtitle
plot.mean
plot.grid
showKhat
col
xlab.size
ylab.size
title.size
legend.size
xtick.distance
left.margin
ytick.distance
col.border

plot_beta_sim

Plot simulated data

Description

Create a plot with the beta sampled from its distribution, as a function of context-level feature $W$. Only works for the hierarchical model (hdpGLM), not the dpGLM

Usage

plot_beta_sim(data, w.idx, ncol = NULL)
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td>the output of the function <code>hdpGLM_simulateData</code></td>
</tr>
<tr>
<td><code>w.idx</code></td>
<td>integer, the index of the context level covariate the plot</td>
</tr>
<tr>
<td><code>ncol</code></td>
<td>integer, the number of columns in the grid of the plot</td>
</tr>
</tbody>
</table>

Description

This function creates a plot with two grids. One is the grid with posterior expectation of betas as function of context-level covariates. The other is the posterior distribution of tau.

Usage

```r
plot_hdpglm(
  samples,
  X = NULL,
  W = NULL,
  ncol.taus = 1,
  ncol.betas = NULL,
  ncol.w = NULL,
  nrow.w = NULL,
  smooth.line = FALSE,
  pred.pexp.beta = FALSE,
  title.tau = NULL,
  true.tau = NULL,
  title.beta = NULL,
  tau.x.axis.size = 1.1,
  tau.y.axis.size = 1.1,
  tau.title.size = 1.2,
  tau.panel.title.size = 1.4,
  tau.legend.size = 1,
  beta.x.axis.size = 1.1,
  beta.y.axis.size = 1.1,
  beta.title.size = 1.2,
  beta.panel.title.size = 1.4,
  beta.legend.size = 1,
  tau.xlab = NULL
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>samples</code></td>
<td>an output of the function <code>hdpGLM</code></td>
</tr>
<tr>
<td><code>X</code></td>
<td>a string vector with the name of the first-level covariates whose associated tau should be displayed</td>
</tr>
</tbody>
</table>
\texttt{w} \hspace{1cm} \text{a string vector with the name of the context-level covariate(s) whose linear effect will be displayed. If \texttt{NULL}, the linear effect tau of all context-level covariates are displayed. Note: the context-level covariate must have been included in the estimation of the model.}

\texttt{ncol.taus} \hspace{1cm} \text{integer with the number of columns of the grid containing the posterior distribution of tau}

\texttt{ncol.betas} \hspace{1cm} \text{integer with the number of columns of the posterior expectation of betas as function of context-level features}

\texttt{ncol.w} \hspace{1cm} \text{integer with the number of columns to use to display the different context-level covariates}

\texttt{nrow.w} \hspace{1cm} \text{integer with the number of rows to use to display the different context-level covariates}

\texttt{smooth.line} \hspace{1cm} \text{boolean, if \texttt{TRUE} the plot will display a regression line representing the regression of the posterior expectation of the linear coefficients betas on the context-level covariates. Default \texttt{FALSE}}

\texttt{pred.pexp.beta} \hspace{1cm} \text{boolean, if \texttt{TRUE} the plots will display a line with the predicted posterior expectation of betas obtained using the posterior expectation of taus, the linear coefficients of the expectation of beta}

\texttt{title.tau} \hspace{1cm} \text{string, the title for the posterior distribution of the context effects}

\texttt{true.tau} \hspace{1cm} \text{a \texttt{data.frame} with four columns. The first must be named \texttt{w} and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named \texttt{beta} must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named Parameter must be named \texttt{tau<w><beta>}, where \texttt{w} and \texttt{beta} must be the actual values displayed in the columns \texttt{w} and \texttt{beta}. Finally, it must have a column named \texttt{True} with the true value of the parameter.}

\texttt{title.beta} \hspace{1cm} \text{string, the title for the posterior expectation of beta as function of context-level covariate}

\texttt{tau.x.axis.size} \hspace{1cm} \text{numeric, relative size of the x-axis of the plot with tau}

\texttt{tau.y.axis.size} \hspace{1cm} \text{numeric, relative size of the y-axis of the plot with tau}

\texttt{tau.title.size} \hspace{1cm} \text{numeric, relative size of the title of the plot with tau}

\texttt{tau.panel.title.size} \hspace{1cm} \text{numeric, relative size of the title of the panels of the plot with tau}

\texttt{tau.legend.size} \hspace{1cm} \text{numeric, relative size of the legend of the plot with tau}

\texttt{beta.x.axis.size} \hspace{1cm} \text{numeric, relative size of the x-axis of the plot with beta}

\texttt{beta.y.axis.size} \hspace{1cm} \text{numeric, relative size of the y-axis of the plot with beta}

\texttt{beta.title.size} \hspace{1cm} \text{numeric, relative size of the title of the plot with beta}

\texttt{beta.panel.title.size} \hspace{1cm} \text{numeric, relative size of the title of the panels of the plot with beta}
beta.legend.size
numeric, relative size of the legend of the plot with beta

tau.xlab
string, the label of the x-axis for the plot with tau

Examples

library(magrittr)
# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
  x2 = rnorm(n, 3),
  z = sample(1:3, n, replace=TRUE),
  y =I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
   I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
   I(z==3) * (3 - 4*x1 - x2 + rnorm(n)),
  w = 20)
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
  x2 = rnorm(n, 3),
  z = sample(1:2, n, replace=TRUE),
  y =I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
   I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
  w = 10)
data = data.context1 %>%
dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_hdpGLM(samples)
plot_hdpGLM(samples, ncol.taus=2, ncol.betas=2, X='x1')
plot_hdpGLM(samples, ncol.taus=2, ncol.betas=2, X='x1', ncol.w=2, nrow.w=1,
  pred.pexp.beta=TRUE,smooth.line=TRUE )

---

**plot_pexp_beta**

Plot beta posterior expectation

**Description**

This function plots the posterior expectation of beta, the linear effect of the individual level covariates, as function of the context-level covariates.
Usage

plot_pexp_beta(
  samples,
  X = NULL,
  W = NULL,
  pred.pexp.beta = FALSE,
  ncol.beta = NULL,
  ylab = NULL,
  nrow.w = NULL,
  ncol.w = NULL,
  smooth.line = FALSE,
  title = NULL,
  legend.position = "top",
  col.pred.line = "red",
  x.axis.size = 1.1,
  y.axis.size = 1.1,
  title.size = 12,
  panel.title.size = 1.4,
  legend.size = 1
)

Arguments

samples  an output of the function hdpGLM
X         a string vector with the name of the first-level covariates whose associated tau
          should be displayed
W         a string vector with the name of the context-level covariate(s) whose linear effect
          will be displayed. If NULL, the linear effect tau of all context-level covariates are
          displayed. Note: the context-level covariate must have been included in the
          estimation of the model.
pred.pexp.beta boolean, if TRUE the plots will display a line with the predicted posterior ex-
          pectation of betas obtained using the posterior expectation of taus, the linear
          coefficients of the expectation of beta
ncol.beta integer with number of columns of the grid used for each group of context-level
          covariates
ylab      string, the label of the y-axis
nrow.w    integer with the number of rows of the grid
ncol.w    integer with the number of columns of the grid
smooth.line boolean, if TRUE the plot will display a regression line representing the regres-
          sion of the posterior expectation of the linear coefficients betas on the context-
          level covariates. Default FALSE
title     string, title of the plot
legend.position one of four options: "bottom" (default), "top", "left", or "right". It indicates the
                position of the legend
col.pred.line string with color of fitted line. Only works if pred.pexp.beta=TRUE
x.axis.size numeric, the relative size of the label in the x-axis
y.axis.size numeric, the relative size of the label in the y-axis
title.size numeric, the absolute size of the title
panel.title.size numeric, the relative size of the titles in the panel of the plot
legend.size numeric, the relative size of the legend

Examples

```r
library(magrittr)
set.seed(66)

# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
                              x2 = rnorm(n, 3),
                              z = sample(1:3, n, replace=TRUE),
                              y = I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                                 I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                                 I(z==3) * (3 - 4*x1 - x2 + rnorm(n)),
                              w = 20)
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
                              x2 = rnorm(n, 3),
                              z = sample(1:2, n, replace=TRUE),
                              y = I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
                                 I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
                              w = 10)
data = data.context1 %>%
              dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_pexp_beta(samples)
plot_pexp_beta(samples, X='x1', ncol.w=2, nrow.w=1)
plot_pexp_beta(samples, X='x1', ncol.beta=2)
plot_pexp_beta(samples, pred.pexp.beta=TRUE, W="w", X=c("x1", "x2"))
plot_pexp_beta(samples, W="w", smooth.line=TRUE, pred.pexp.beta=TRUE, ncol.beta=2)
```
**plot_tau**

**Plot tau**

**Description**

Function to plot posterior distribution of tau

**Usage**

```r
plot_tau(
  samples,
  X = NULL,
  W = NULL,
  title = NULL,
  true.tau = NULL,
  show.all.taus = FALSE,
  show.all.betas = FALSE,
  ncol = NULL,
  legend.position = "top",
  x.axis.size = 1.1,
  y.axis.size = 1.1,
  title.size = 1.2,
  panel.title.size = 1.4,
  legend.size = 1,
  xlab = NULL
)
```

**Arguments**

- `samples`: an output of the function `hdpGLM`
- `X`: a string vector with the name of the first-level covariates whose associated tau should be displayed
- `W`: a string vector with the name of the context-level covariate(s) whose linear effect will be displayed. If `NULL`, the linear effect tau of all context-level covariates are displayed. Note: the context-level covariate must have been included in the estimation of the model.
- `title`: string, title of the plot
- `true.tau`: a `data.frame` with four columns. The first must be named `w` and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named `beta` must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named `Parameter` must be named `tau<w><beta>`, where `w` and `beta` must be the actual values displayed in the columns `w` and `beta`. Finally, it must have a column named `True` with the true value of the parameter.
- `show.all.taus`: boolean, if FALSE (default) the posterior distribution of taus representing the intercept of the expectation of beta are omitted
show.all.betas  boolean, if FALSE (default) the taus affecting only the intercept terms of the outcome variable are omitted
ncol    number of columns of the grid. If NULL, one column is used
legend.position    one of four options: "bottom" (default), "top", "left", or "right". It indicates the position of the legend
x.axis.size numeric, the relative size of the label in the x-axis
y.axis.size numeric, the relative size of the label in the y-axis
title.size numeric, the relative size of the title of the plot
panel.title.size numeric, the relative size of the titles in the panel of the plot
legend.size numeric, the relative size of the legend
xlab    string, the label of the x-axis

Examples

library(magrittr)
set.seed(66)

# Note: this example is just for illustration. MCMC iterations are very reduced
set.seed(10)
n = 20
data.context1 = tibble::tibble(x1 = rnorm(n, -3),
                                x2 = rnorm(n, 3),
                                z = sample(1:3, n, replace=TRUE),
                                y =I(z==1) * (3 + 4*x1 - x2 + rnorm(n)) +
                                I(z==2) * (3 + 2*x1 + x2 + rnorm(n)) +
                                I(z==3) * (3 - 4*x1 - x2 + rnorm(n)),
                                w = 20)
data.context2 = tibble::tibble(x1 = rnorm(n, -3),
                                x2 = rnorm(n, 3),
                                z = sample(1:2, n, replace=TRUE),
                                y =I(z==1) * (1 + 3*x1 - 2*x2 + rnorm(n)) +
                                I(z==2) * (1 - 2*x1 + x2 + rnorm(n)),
                                w = 10)
data = data.context1 %>%
  dplyr::bind_rows(data.context2)

## estimation
mcmc = list(burn.in=1, n.iter=50)
samples = hdpGLM(y ~ x1 + x2, y ~ w, data=data, mcmc=mcmc, n.display=1)

plot_tau(samples)
plot_tau(samples, ncol=2)
plot_tau(samples, X='x1', W='w')
predict.dpGLM

plot_tau(samples, show.all.taus=TRUE, show.all.betas=TRUE, ncol=2)

predict.dpGLM | dpGLM Predicted values

Description

Function returns the predicted (fitted) values of the outcome variable using the estimated posterior expectation of the linear covariate betas produced by the hdpGLM function.

Usage

```r
## S3 method for class 'dpGLM'
predict(object, new_data = NULL, ...)
```

Arguments

- `object`: outcome of the function hdpGLM
- `new_data`: data frame with the values of the covariates that are going to be used to generate the predicted/fitted values. The posterior mean is used to create the predicted values.
- `...`: family: a string with the family of the output variable: gaussian (default), binomial, etc...

Value

It returns a data.frame with the fitted values for the outcome variable, which are produced using the estimated posterior expectation of the linear coefficients beta.

predict.hdpGLM | hdpGLM Predicted values

Description

Function returns the predicted (fitted) values of the outcome variable using the estimated posterior expectation of the linear covariate betas produced by the hdpGLM function.

Usage

```r
## S3 method for class 'hdpGLM'
predict(object, new_data = NULL, ...)
```
print.dpGLM

Arguments

- **object**: outcome of the function hdpLGM
- **new_data**: data frame with the values of the covariates that are going to be used to generate the predicted/fitted values. The posterior mean is used to create the predicted values
- **...**
  - **family**: a string with the family of the output variable: `gaussian` (default), `binomial`, etc...

Value

It returns a data.frame with the fitted values for the outcome variable, which are produced using the estimated posterior expectation of the linear coefficients beta.

---

print.dpGLM

Description

Generic method to print the output of the dpGLM function

Usage

```r
## S3 method for class 'dpGLM'
print(x, ...)
```

Arguments

- **x**: a dpGLM object returned by the function hdpGLM
- **...**: ignore

Value

returns a summary of the posterior distribution of the parameters
**print.dpGLM_data**

---

**Description**

Generic method to print the output of the `hdpGLM_simulateData` function

**Usage**

```r
## S3 method for class 'dpGLM_data'
print(x, ..., )
```

**Arguments**

- `x` a `dpGLM_data` object returned by the function `hdpGLM_simulateData`
- `...` ignore

**Value**

returns a summary of the simulated data

---

**print.hdpGLM**

---

**Description**

Generic method to print the output of the `hdpGLM` function

**Usage**

```r
## S3 method for class 'hdpGLM'
print(x, ..., )
```

**Arguments**

- `x` a `hdpGLM` object returned by the function `hdpGLM`
- `...` ignore

**Value**

returns a summary of the posterior distribution of the parameters
print.hdpGLM_data

Description

Generic method to print the output of the hdpGLM_simulateData function

Usage

## S3 method for class 'hdpGLM_data'
print(x, ...)

Arguments

x        a hdpGLM_data object returned by the function hdpGLM_simulateData
...      ignore

Value

returns a summary of the simulated data

summary.dpGLM

Description

This function provides a summary of the MCMC samples from the dpGLM model

Usage

## S3 method for class 'dpGLM'
summary(object, ...)

Arguments

object    a dpGLM object returned by the function hdpGLM
...       The additional parameters accepted are:
           true.beta: (see plot.dpGLM)

Details

Data points are assigned to clusters according to the highest estimated probability of belonging to that cluster
Summary dpGLM data

**Description**

This function summarizes the data and parameters used to generate the data using the function hdpLGM.

**Usage**

```r
## S3 method for class 'dpGLM_data'
summary(object, ...)
```

**Arguments**

- `object` an object of the class dpGLM_data
- `...` ignored

**Value**

The function returns a list with the summary of the data produced by the standard summary function and a data.frame with the true values of beta for each cluster.

Summary for hdpGLM class

**Description**

This is a generic summary function that describes the output of the function hdpGLM.

**Usage**

```r
## S3 method for class 'hdpGLM'
summary(object, ...)
```

**Arguments**

- `object` an object of the class hdpGLM generated by the function hdpGLM
- `...` Additional arguments accepted are:
  - `true.beta`: a data.frame with the true values of the linear coefficients beta if they are known. The data.frame must contain a column named `j` with the index of the context associated with that particular linear coefficient beta. It must match the indexes used in the data set for each context. Another column named `k` must be provided, indicating the cluster of beta, and a column named `Parameter` with the name of the linear coefficients (beta1, beta2, ..., beta_dx, ...).
where $dx$ is the number of covariates at the individual level, and beta1 is the coefficient of the intercept term). It must contain a column named `True` with the true value of the betas. Finally, the `data.frame` must contain columns with the context-level covariates as used in the estimation of the `hdpGLM` function (see `Details` below).

`true.tau`: a `data.frame` with four columns. The first must be named `w` and it indicates the index of each context-level covariate, starting with 0 for the intercept term. The second column named `beta` must contain the indexes of the betas of individual-level covariates, starting with 0 for the intercept term. The third column named `Parameter` must be named `tau<><beta>`, where `w` and `beta` must be the actual values displayed in the columns `w` and `beta`. Finally, it must have a column named `True` with the true value of the parameter.

**Details**

The function `hdpGLM` returns a list with the samples from the posterior distribution along with other elements. That list contains an element named `context.cov` that connects the indexed "C" created during the estimation and the context-level covariates. So each unique context-level covariate gets an index during the estimation. The algorithm only requires the context-level covariates, but it creates such index C to help the estimation. If `true.beta` is provided, it must contain indexes for the context as well, which indicates the context of each specific linear coefficient `beta`. Such index will probably be different from the one created by the algorithm. Therefore, when the `true.beta` is provided, we need to connect the context index C generated by the algorithm and the column j in the `true.beta` data.frame in order to compare the true values and the estimated value for each context. That is why we need the values of the context-level covariates as well. The summary uses them as key to merge the true and the estimated values for each context. The true and estimated clusters are matched based on the shortest distance between the estimated posterior average and the true value in each context because the labels of the clusters in the estimation can vary, even thought the same data points are classified in the same clusters.

**Value**

The function returns a list with two `data.frame`s. The first summarizes the posterior distribution of the linear coefficients `beta`. The mean, median, and the 95% HPD interval are provided. The second `data.frame` contains the summary of the posterior distribution of the parameter `tau`.

---

**summary.hdpGLM_data**

**Summary**

**Description**

This functions summarizes the data simulated by the function `hdpGLM_simulateData`.

**Usage**

```r
## S3 method for class 'hdpGLM_data'
summary(object, ...)
```
**summary_tidy**

**Arguments**

- **object** an object of the class `hdpGLM_data`, which is produced by the function `hdpGLM_simulateData`
- ... ignored

**Value**

It returns a list with three elements. The first is a summary of the data, the second a tibble with the linear coefficients beta and their values used to generate the data, and the third element is also a tibble with the true values of tau used to generate the betas.

**Description**

This function provides a summary of the MCMC samples from the dpGLM model

**Usage**

```r
summary_tidy(object, ...)
```

**Arguments**

- **object** a dpGLM object returned by the function `hdpGLM`
- ... The additional parameters accepted are:
  - true.beta: (see `plot.dpGLM`)

**Details**

Data points are assigned to clusters according to the highest estimated probability of belonging to that cluster

**welfare**

**Fake data set with 2000 observations**

**Description**

A dataset containing simulated data about public opinion

**Usage**

```r
welfare
```
Format

A data frame with 2000 rows and 4 variables:

- **support** support for welfare policies
- **inequality** levels of inequality in the neighborhood
- **income** individual-level income
- **ideology** individual-level ideology

Source

Simulated data

---

**welfare2**

*Fake data set with 2000 observations*

Description

A dataset containing simulated data about public opinion in different countries

Usage

welfare2

Format

A data frame with 2000 rows and 6 variables:

- **support** support for welfare policies
- **inequality** levels of inequality in the neighborhood
- **income** individual-level income
- **ideology** individual-level ideology
- **country** country label or index
- **gap** country-level gender gap in country’s provision of public good

Source

Simulated data
Index

* datasets
  welfare, 31
  welfare2, 32

classify, 2
coefficient.dpGLM, 3
coefficient.hdpGLM, 3

hdpGLM, 2, 3, 4, 7, 12, 14–18, 21, 23, 29, 30
hdpGLM_classify, 6
hdpGLM_package, 7
hdpGLM_simParameters, 8
hdpGLM_simulateData, 9

lm, 4

mcmc_info.dpGLM, 11
mcmc_info.hdpGLM, 11

nclusters, 11

plot.dpGLM, 12, 28, 31
plot.hdpGLM, 14
plot_beta, 16
plot_beta_sim, 17
plot_hdpGLM, 18
plot_pexp_beta, 20
plot_tau, 23
predict.dpGLM, 25
predict.hdpGLM, 25
print.dpGLM, 26
print.dpGLM_data, 27
print.hdpGLM, 27
print.hdpGLM_data, 28

summary.dpGLM, 28
summary.dpGLM_data, 29
summary.hdpGLM, 29
summary.hdpGLM_data, 30
summary_tidy, 31

welfare, 31
welfare2, 32