Package: gpss (via r-universe)

October 26, 2026

Title Gaussian Processes for Social Science

Version 1.0.3

Date 2026-04-01

Description Provides Gaussian process (GP) regression tools for social science inference problems. GPs combine flexible nonparametric regression with principled uncertainty quantification: rather than committing to a single model fit, the posterior reflects lesser knowledge at the edge of or beyond the observed data, where other approaches become highly model-dependent. The package reduces user-chosen hyperparameters from three to zero and supplies convenience functions for regression discontinuity (gp_rdd()), interrupted time-series (gp_its()), and general GP fitting (gpss(), gp_train(), gp_predict()). Methods are described in Cho, Kim, and Hazlett (2026) <doi:10.1017/pan.2026.10032>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

URL https://doeunkim.org/gpss/, https://github.com/doeun-kim/gpss

BugReports https://github.com/doeun-kim/gpss/issues

Imports Rcpp (>= 1.0.14), Matrix, ggplot2, rlang

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5.0)

LazyData true

Suggests MASS, posterior, testthat (>= 3.0.0)

NeedsCompilation yes

Author Soonhong Cho [aut], Doeun Kim [aut] (ORCID: <https://orcid.org/0000-0003-4789-6599>), Chad Hazlett [aut, cre]

Maintainer Chad Hazlett <chazlett@ucla.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-04-08 21:05:31 UTC

RemoteUrl https://github.com/cran/gpss

RemoteRef HEAD

RemoteSha db13ce6c593258e1bd2be42ebcb52c03ff992b5d

Contents

gp_its 2
gp_predict 4
gp_rdd 5
gp_rdd_plot 6
gp_train 7
lalonde 9
predict.gpss 10

Index **12**

`gp_its` *Gaussian Process Interrupted Time Series*

Description

Gaussian Process Interrupted Time Series

Usage

```
gp_its(
  y,
  dates,
  date_treat,
  covariates = NULL,
  kernel_type = "gaussian",
  b = NULL,
  s2 = 0.3,
  period = NULL,
  auto_period = FALSE,
  time_col = NULL,
  scale = TRUE,
  optimize = TRUE,
  interval_type = c("prediction", "confidence"),
  alpha = 0.05,
  mixed_data = FALSE,
  cat_columns = NULL,
  placebo_check = FALSE,
  placebo_periods = NULL,
```

```
  verbose = FALSE
)
```

Arguments

y	numeric vector or ts object of outcome values
dates	Date vector of observation dates
date_treat	Date of treatment/intervention
covariates	optional matrix/data.frame of covariates
kernel_type	kernel type for GP
b	bandwidth (NULL for automatic)
s2	noise variance (default 0.3)
period	period for periodic kernels
auto_period	logical; auto-detect period from pre-treatment data
time_col	character or numeric; which column is the time variable
scale	logical; scale covariates
optimize	logical; optimize s2 via MLE
interval_type	"prediction" or "confidence"
alpha	significance level
mixed_data	logical; whether covariates contain categorical variables
cat_columns	character vector of categorical column names
placebo_check	logical; run placebo checks
placebo_periods	number of placebo periods (NULL for automatic)
verbose	logical; print progress messages

Value

An object of class "gp_its", a list containing:

y	the outcome vector
dates	the date vector
date_treat	the treatment date
y0_hat	counterfactual predictions for post-treatment periods
estimates	data frame with columns tau_t, tau_cum, tau_avg
se	data frame of standard errors
ci	data frame of confidence interval bounds
gp_model	the fitted GP model from gp_train
placebo_estimates	placebo check results (if requested)

Examples

```
# Simulated interrupted time series
set.seed(42)
n <- 50
dates <- seq(as.Date("2020-01-01"), by = "month", length.out = n)
y <- rnorm(n) + c(rep(0, 30), rep(2, 20))
res <- gp_its(y, dates, date_treat = as.Date("2022-07-01"))
print(res)
```

`gp_predict` *gp_predict*

Description

to predict outcome values at testing points by feeding the results obtained from gp_train()

Usage

```
gp_predict(gp, Xtest, prior_mean = NULL)
```

Arguments

`gp`	a list-form object obtained from gp_train()
`Xtest`	a data frame or a matrix of testing data set
`prior_mean`	a numeric vector of prior mean values for Y at each test point. Required when the model was trained with a `prior_mean`; added to `Ys_mean_orig` to recover predictions on the original Y scale. Must be the same length as `nrow(Xtest)`. (default = NULL)

Value

`Xtest_scaled`	testing data in a scaled form
`Xtest`	the original testing data set
`Ys_mean_scaled`	the predicted values of Y in a scaled form
`Ys_mean_orig`	the predicted values of Y in the original scale
`Ys_cov_scaled`	covariance of predicted Y in a scaled form
`Ys_cov_orig`	covariance of predicted Y in the original scale
`f_cov_orig`	covariance of target function in the original scale
`b`	the bandwidth value obtained from gp_train()
`s2`	the s2 value obtained from gp_train()

Examples

```
data(lalonde)
cat_vars <- c("race_ethnicity", "married")
all_vars <- c("age","educ","re74","re75","married", "race_ethnicity")

X <- lalonde[,all_vars]
Y <- lalonde$re78
D <- lalonde$nsw

X_train <- X[D==0,]
Y_train <- Y[D==0]
X_test <- X[D == 1,]
Y_test <- Y[D == 1]

gp_train.out <- gp_train(X = X_train, Y = Y_train,
optimize=TRUE, mixed_data = TRUE,
cat_columns = cat_vars)
gp_predict.out <- gp_predict(gp_train.out, X_test)
```

---

gp_rdd *gp_rdd*

Description

This function implements a regression discontinuity (RD) design using Gaussian Process (GP) regression on each side of the cutoff. Separate GP models are estimated for observations below and above the threshold, allowing for flexible and fully nonparametric functional forms. The treatment effect is computed as the difference between the predicted conditional means at the cutoff from the right and left limits. Standard errors and confidence intervals are constructed using the posterior predictive variance from the two independently fitted GPs.

Usage

```
gp_rdd(
  X,
  Y,
  cut,
  alpha = 0.05,
  b = NULL,
  trim = FALSE,
  trim_k_value = 0.1,
  scale = TRUE
)
```

Arguments

X	forcing variable
Y	Y vector (outcome variable)
cut	cut point
alpha	confidence level (default = 0.05)
b	bandwidth (default = NULL)
trim	a logical value indicating whether you want to do an automatic trim at a specific value of trim_k_value (default=FALSE)
trim_k_value	a numerical value indicating the kernel value that you want to trim above (default = 0.1)
scale	a logical value indicating whether you want to scale the covariates (default = TRUE)

Value

tau	an estimated treatment effect
se	the standard error of tau

Examples

```
n <- 100
tau <- 3
cut <- 0
x <- rnorm(n, 0, 1)
y <- rnorm(n, 0, 1) + tau*ifelse(x>cut, 1, 0)
gp_rdd.out <- gp_rdd(x, y, cut)
gp_rdd_plot(gp_rdd.out)
```

---

gp_rdd_plot *gp_rdd_plot*

Description

to draw an RD plot using the results obtained from gp_rdd()

Usage

```
gp_rdd_plot(gp_rdd_res, l_col = "blue", r_col = "red")
```

Arguments

gp_rdd_res	a list-form results obtained from gp_rdd()
l_col	a character value indicating the color of the left side of the cutoff point (default = "blue")
r_col	a character value indicating the color of the right side of the cutoff point (default = "red")

Value

A ggplot object showing the RD plot.

Examples

```
library(ggplot2)
n <- 100
tau <- 3
cut <- 0
x <- rnorm(n, 0, 1)
y <- rnorm(n, 0, 1) + tau*ifelse(x>cut, 1, 0)
gp_rdd.out <- gp_rdd(x, y, cut)
gp_rdd_plot(gp_rdd.out) +
 geom_vline(xintercept = cut, lty="dashed")
```

---

gp_train *gp_train*

Description

to train GP model with training data set

Usage

```
gp_train(
  X,
  Y,
  b = NULL,
  s2 = 0.3,
  optimize = FALSE,
  scale = TRUE,
  kernel_type = "gaussian",
  period = NULL,
  time_col = NULL,
  mixed_data = FALSE,
  cat_columns = NULL,
  Xtest = NULL,
  prior_mean = NULL
)
```

Arguments

X	a set of covariate data frame or matrix
Y	Y vector (outcome variable)
b	bandwidth (default = NULL)
s2	noise or a fraction of Y not explained by X (default = 0.3)

optimize	a logical value to indicate whether an automatic optimized value of S2 should be used. If FALSE, users must define s2. (default = FALSE)
scale	a logical value to indicate whether covariates should be scaled. (default = TRUE)
kernel_type	a character value indicating the kernel type (default = "gaussian")
period	a numeric value for the period parameter, required for periodic kernels (default = NULL)
time_col	a character or numeric value indicating which column is the time variable. If specified, this column will be moved to the first position for correct period scaling in periodic kernels. (default = NULL)
mixed_data	a logical value to indicate whether the covariates contain a categorical/binary variable (default = FALSE)
cat_columns	a character or a numerical vector indicating categorical variables. Must be character (not numeric) when time_col is specified. (default = NULL)
Xtest	a data frame or a matrix of testing covariates. This is necessary when a non-overlapping categorical value exists between training and testing data sets. (default = NULL)
prior_mean	a numeric vector of prior mean values for Y at each training observation. If provided, the GP is fitted to the residuals (Y - prior_mean), and the prior mean is added back to recover predictions on the original Y scale. Must be the same length as Y. (default = NULL)

Value

post_mean_scaled	posterior distribution of Y in a scaled form
post_mean_orig	posterior distribution of Y in an original scale
post_cov_scaled	posterior covariance matrix in a scaled form
post_cov_orig	posterior covariance matrix in an original scale
K	a kernel matrix of X
prior_mean_scaled	prior distribution of mean in a scaled form
X.orig	the original matrix or data set of X
X.init	the original matrix or data set of X with categorical variables in an expanded form
X.init.mean	the initial mean values of X
X.init.sd	the initial standard deviation values of X
Y.init.mean	the initial mean value of Y
Y.init.sd	the initial standard deviation value of Y
K	the kernel matrix of X
Y	scaled Y
X	scaled X

b	bandwidth
s2	sigma squared
alpha	alpha value in Rasmussen and Williams (2006) p.19
L	L value in Rasmussen and Williams (2006) p.19
mixed_data	a logical value indicating whether X contains a categorical/binary variable
cat_columns	a character or a numerical vector indicating the location of categorical/binary variables in X
cat_num	a numerical vector indicating the location of categorical/binary variables in an expanded version of X
time_col	the time column specification used (or NULL if not specified)
Xcolnames	column names of X
prior_mean	the prior mean vector supplied at training (or NULL)

Examples

```
data(lalonde)
cat_vars <- c("race_ethnicity", "married")
all_vars <- c("age","educ","re74","re75","married", "race_ethnicity")

X <- lalonde[,all_vars]
Y <- lalonde$re78
D <- lalonde$nsw

X_train <- X[D==0,]
Y_train <- Y[D==0]
X_test <- X[D == 1,]
Y_test <- Y[D == 1]

gp_train.out <- gp_train(X = X_train, Y = Y_train,
optimize=TRUE, mixed_data = TRUE,
cat_columns = cat_vars)
gp_predict.out <- gp_predict(gp_train.out, X_test)
```

---

lalonde *Data from National Supported Work program and Panel Study in Income Dynamics*

Description

Dehejia and Wahba (1999) sample of data from Lalonde (1986). This data set includes 185 treated units from the National Supported Work (NSW) program, paired with 2490 control units drawn from the Panel Study of Income Dynamics (PSID-1).

The treatment variable of interest is `nsw`, which indicates that an individual was in the job training program. The main outcome of interest is real earnings in 1978 (`re78`). The remaining variables are characteristics of the individuals, to be used as controls.

Usage

```
lalonde
```

Format

A data frame with 2675 rows and 14 columns.

nsw treatment indicator: participation in the National Supported Work program.

re78 real earnings in 1978 (outcome)

u78 unemployed in 1978; actually an indicator for zero income in 1978

age age in years

black indicator for identifying as black

hisp indicator for identifying as Hispanic

race_ethnicity factor for self-identified race/ethnicity; same information as `black` and `hisp` in character form.

married indicator for being married

re74 real income in 1974

re75 real income in 1975

u74 unemployment in 1974; actually an indicator for zero income in 1974

u75 unemployment in 1975; actually an indicator for zero income in 1975

educ Years of education of the individual

nodegr indicator for no high school degree; actually an indicator for years of education less than 12

References

Dehejia, Rajeev H., and Sadek Wahba. "Causal effects in non-experimental studies: Reevaluating the evaluation of training programs." Journal of the American statistical Association 94.448 (1999): 1053-1062.

LaLonde, Robert J. "Evaluating the econometric evaluations of training programs with experimental data." The American economic review (1986): 604-620.

`predict.gpss` *predict method for* `gpss` *objects*

Description

predict method for `gpss` objects

Usage

```
## S3 method for class 'gpss'
predict(
  object,
  newdata = NULL,
  type = "response",
  format = "default",
  interval = "confidence",
  level = 0.95,
  prior_mean = NULL,
  ...
)
```

Arguments

object	a model object for which prediction is desired.
newdata	data frame on which to make predictions (test set)
type	"response" or "scaled"
format	"default" or "rvar"
interval	"prediction" or "confidence"
level	a numerical value between 0 and 1
prior_mean	a numeric vector of prior mean values for Y at each test observation. Required when the model was trained with a prior_mean; see gp_predict. (default = NULL)
...	additional arguments (not used)

Value

If format = "default", a numeric matrix with columns fit, lwr, and upr giving the posterior mean and the lower and upper bounds of the requested interval for each row of newdata. If format = "rvar", a copy of newdata with an additional column rvar containing a posterior random variable (class rvar from the **posterior** package).

Examples

```
library(gpss)
data(lalonde)
# categorical variables must be encoded as factors
dat <- transform(lalonde, race_ethnicity = factor(race_ethnicity))
# train and test sets
idx <- sample(seq_len(nrow(dat)), 500)
dat_train <- dat[idx, ]
dat_test <- dat[-idx, ]
# Fit model
mod <- gpss(re78 ~ nsw + age + educ + race_ethnicity, data = dat_train)
p <- predict(mod, newdata = dat_test)
p_confidence99 <- predict(mod, newdata = dat_test, interval = "confidence", level = 0.99)
```

Index

∗ **datasets**
lalonde, 9

gp_its, 2
gp_predict, 4, *11*
gp_rdd, 5
gp_rdd_plot, 6
gp_train, *3*, 7

lalonde, 9

predict.gpss, 10