Package: BSCB (via r-universe)

July 11, 2026

Title Bayesian Simultaneous Credible Bands for Polynomial Regression

Version 1.0.0

Description Provides functions to construct two-sided Bayesian simultaneous credible bands (BSCBs) for the regression curve in univariate polynomial regression over a finite covariate interval. Six methods are implemented, including Normal-Gamma conjugate priors (with empirical Bayes, unit-information, and g-prior hyperparameter specifications), non-conjugate priors fitted via Hamiltonian Monte Carlo (HMC) using 'cmdstanr', and a non-informative independent Jeffreys prior approach. Also includes functions for computing the empirical simultaneous coverage rate (ESCR) and posterior simultaneous coverage probability (PSCP), enabling performance comparison across methods. The methodology is described in: Yang, F., Han, Y., Liu, W., & Hall, I. (2026). ``Bayesian simultaneous credible bands for polynomial regression'' <doi:10.48550/arXiv.2606.28015>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests DEoptim, testthat (>= 3.0.0), cmdstanr, knitr, rmarkdown, ggplot2

VignetteBuilder knitr

Additional_repositories `https://stan-dev.r-universe.dev`

Imports mvtnorm, MASS, OptimalDesign, instantiate, posterior

Config/testthat/edition 3

Depends R (>= 4.0)

Date 2026-07-02

URL `https://github.com/fannyyang73/BSCB`

Config/instantiate/stan-dir src/stan

BugReports https://github.com/fannyyang73/BSCB/issues

Repository https://fannyyang73.r-universe.dev

Date/Publication 2026-07-11 09:12:09 UTC

RemoteUrl https://github.com/fannyyang73/bscb

RemoteRef HEAD

RemoteSha b9620c75f1adc5485d51b75ca26469fcab1d5caa

Contents

`compute_bpcb_ind_jeffreys` *BPCB-I-J: the Bayesian pointwise credible band using the independent Jeffreys prior*

Description

Constructs a $(1-\alpha)$ two-sided Bayesian pointwise credible band (BPCB) for polynomial regression using the independent Jeffreys prior. Unlike the simultaneous credible band, the critical constant λ is derived analytically from the marginal t-distribution as $t_{n-p-1}^{\alpha/2}$.

Usage

```
compute_bpcb_ind_jeffreys(
  X,
  Y,
  alpha = 0.05,
  a = NULL,
  b = NULL,
  L = 50000,
  AR_setting = 0,
  rho = NULL,
  theta_true = NULL,
  verbose = TRUE
)
```

Arguments

`X`	Numeric matrix of dimension $n \times (p+1)$. Design matrix with intercept in the first column.
`Y`	Numeric vector of length n. Response variable.
`alpha`	Numeric. Nominal mis-coverage level; the band targets $1-\alpha$ pointwise coverage. Default is `0.05`.
`a`	Numeric. Left endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
`b`	Numeric. Right endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
`L`	Integer. Not used in this function (included for API consistency with other `compute_bscb_*` functions). Default is `500000`.
`AR_setting`	Integer. Error covariance structure: `0` = i.i.d. errors (default); `1` = AR(1) errors.
`rho`	Numeric. AR(1) coefficient. Required when `AR_setting` = `1`.
`theta_true`	Numeric vector of length $p+1$. True regression coefficients. Optional; stored in the output for diagnostic use.
`verbose`	Logical. If `TRUE` (default), prints the value of the critical constant lambda.

Value

An object of class `"bpcb_fit"`, a list containing:

lambda Critical constant $t^{\alpha/2}_{n-p-1}$ for the credible band.

lower_bound Function: computes the lower band at a given `x`.

upper_bound Function: computes the upper band at a given `x`.

mu_star Posterior mean of θ (GLS estimate).

dof Degrees of freedom of the marginal posterior ($n-p-1$).

scale_mat Scale matrix Σ_0 of the marginal multivariate-t posterior distribution of θ.

cov_theta Posterior covariance matrix of θ. The posterior covariance matrix equals $\mathrm{Cov}(\theta) = \frac{\nu}{\nu-2}\Sigma_0$, where ν is the degrees of freedom (`dof`).

x_range Covariate domain $[a, b]$.

theta_true True parameters (if supplied).

method Character string `"independent_jeffreys"`.

params List of configuration parameters.

See Also

`compute_bscb_ind_jeffreys` for the simultaneous version, `compute_bscb_conjugate` for the conjugate prior version.

Examples

```
# Quadratic model with i.i.d. errors
set.seed(123)
n <- 50
x <- seq(-5, 5, length.out = n)
X <- cbind(1, x, x^2)
theta_true <- c(-6, -3, 0.25)
Y <- X %*% theta_true + rnorm(n, sd = 0.2)

fit <- compute_bpcb_ind_jeffreys(
  X          = X,
  Y          = Y,
  alpha      = 0.05,
  a          = -5,
  b          =  5,
  theta_true = theta_true,
  verbose    = FALSE
)

# Critical constant (t quantile)
fit$lambda

# Evaluate the band over a grid
x_seq     <- seq(-5, 5, length.out = 200)
lower_vec <- fit$lower_bound(x_seq)
upper_vec <- fit$upper_bound(x_seq)

# Plot
plot(x_seq, lower_vec, type = "l", col = "red", lty = 2, lwd = 2,
     ylim = range(c(lower_vec, upper_vec, Y)),
     xlab = "x", ylab = "y",
     main = "95% Bayesian Pointwise Credible Band (Indep. Jeffreys)")
lines(x_seq, upper_vec, col = "red", lty = 2, lwd = 2)
lines(x_seq, cbind(1, x_seq, x_seq^2) %*% theta_true,
      col = "blue", lwd = 2)
points(x, Y, pch = 16, col = "gray")
legend("topright",
       legend = c("True curve", "Data", "95% BPCB-J"),
```

```
       col    = c("blue", "gray", "red"),
       lty    = c(1, NA, 2),
       pch    = c(NA, 16, NA))
```

---

compute_bscb_conjugate

BSCB-C: Bayesian Simultaneous Credible Band under the Normal-Gamma Conjugate Prior

---

Description

Constructs a $(1 - \alpha)$ two-sided Bayesian simultaneous credible band for polynomial regression using the normal-gamma conjugate prior. The marginal posterior of θ follows a multivariate-t distribution. The critical constant λ is estimated via Monte Carlo sampling.

Usage

```
compute_bscb_conjugate(
  X,
  Y,
  alpha = 0.05,
  a = NULL,
  b = NULL,
  L = 5e+05,
  AR_setting = 0,
  rho = NULL,
  hyperparameter = c("empirical", "unit_info", "g_prior"),
  optimize_type = c("P", "G", "D"),
  theta_true = NULL,
  verbose = TRUE
)
```

Arguments

X	Numeric matrix of dimension $n \times (p + 1)$. Design matrix with intercept in the first column.
Y	Numeric vector of length n. Response variable.
alpha	Numeric. Nominal mis-coverage level; the band targets $1 - \alpha$ simultaneous coverage. Default is `0.05`.
a	Numeric. Left endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
b	Numeric. Right endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
L	Integer. Number of Monte Carlo draws for computing the critical constant λ. Default is `500000`.

`AR_setting` Integer. Error covariance structure: `0` = i.i.d. errors (default); `1` = AR(1) errors.

`rho` Numeric. AR(1) coefficient. Required when `AR_setting` = `1`.

`hyperparameter` Character. Hyperparameter specification for the Normal-Gamma prior: `"empirical"` = empirical Bayes; `"unit_info"` = unit-information prior; `"g_prior"` = Zellner's g-prior.

`optimize_type` Character. Method for computing $\sup_{x \in [a,b]} T(x)$: `"P"` = polyroot analytical method (recommended); `"G"` = global optimisation; `"D"` = differential evolution (DEoptim).

`theta_true` Numeric vector of length $p + 1$. True regression coefficients. Optional; stored in the output for diagnostic use.

`verbose` Logical. If `TRUE` (default), prints progress messages including the value of the critical constant lambda.

Value

An object of class `"bscb_fit"`, a list containing:

lambda Critical constant for the credible band.

lower_bound Function: computes the lower band at a given `x`.

upper_bound Function: computes the upper band at a given `x`.

mu_star Posterior mean of θ.

dof Degrees of freedom of the marginal posterior.

scale_mat Scale matrix Σ_0 of the marginal multivariate-t posterior distribution of θ.

cov_theta Posterior covariance matrix of θ. The posterior covariance matrix equals $\text{Cov}(\theta) = \frac{\nu}{\nu-2}\Sigma_0$, where ν is the degrees of freedom (`dof`).

x_range Covariate domain $[a, b]$.

lambda_samples Monte Carlo samples used to compute λ.

theta_true True parameters (if supplied).

method Character string `"conjugate"`.

params List of configuration parameters.

See Also

`compute_bscb_ind_jeffreys` for the independent Jeffreys prior version, `compute_bpcb_ind_jeffreys` for the pointwise band.

Examples

```
# Example 1: Simple quadratic model with i.i.d. errors

set.seed(123)
n <- 50
p <- 2
x <- seq(-5, 5, length.out = n)
X <- cbind(1, x, x^2)
```

```
theta_true <- c(-6, -3, 0.25)
e_sd <- 0.2

# Generate response
epsilon <- rnorm(n, mean = 0, sd = e_sd)
Y <- X %*% theta_true + epsilon

# Notably, this is a quick example. In theory, L should set to L=500,000 or larger;
fit <- compute_bscb_conjugate(X=X,  Y=Y, alpha = 0.05, a = -5, b = 5, L = 5000,
                              AR_setting = 0, # 0: iid error; 1: autoregressive error
                              rho = NULL,
                              hyperparameter = "empirical",
                              optimize_type = "P",
                              theta_true = theta_true,
                              verbose = FALSE)


# View results
print(fit$lambda)           # Critical value
print(fit$mu_star)          # Posterior mean of theta
print(fit$cov_theta)        # Posterior covariance matrix


# Compute BSCB-C at a specific point
x_new <- 0.5
lower <- fit$lower_bound(x_new)
upper <- fit$upper_bound(x_new)
cat("At x =", x_new, ": [", lower, ",", upper, "]\n")

# Vectorized computation for plotting
x_seq <- seq(-5, 5, length.out = 1000)
lower_vec <- fit$lower_bound(x_seq)
upper_vec <- fit$upper_bound(x_seq)
y_true <- cbind(1, x_seq, x_seq^2) %*% theta_true

# Visualization
plot(x_seq, lower_vec, type = "l", col = "red", lty = 2, lwd = 2,
     ylim = range(c(lower_vec, upper_vec, Y)),
     xlab = "x", ylab = "y",
     main = "95% Bayesian Simultaneous Credible Band (Conjugate Prior)")
lines(x_seq, upper_vec, col = "red", lty = 2, lwd = 2)
lines(x_seq, y_true, col = "blue", lwd = 2)
points(x, Y, pch = 16, col = "gray")
legend("topright",
       legend = c("True curve", "Data", "95% BSCB-C"),
       col = c("blue", "gray", "red"),
       lty = c(1, NA, 2),
       pch = c(NA, 16, NA),
       lwd = 2)
```

compute_bscb_hmc *Compute BSCB via Hamiltonian Monte Carlo*

Description

Constructs a Bayesian Simultaneous Credible Band (BSCB) for polynomial regression under non-conjugate priors using Hamiltonian Monte Carlo (HMC) via Stan. Supports two prior specifications: Normal-Normal and Normal-half-Cauchy. The critical constant lambda is estimated by Monte Carlo, and the Posterior Simultaneous Coverage Probability (PSCP) is also returned.

Usage

```
compute_bscb_hmc(
  Y,
  X,
  V = diag(nrow(X)),
  a,
  b,
  theta_true = NULL,
  alpha = 0.05,
  prior_type = c("normal_half_cauchy", "normal_normal"),
  normal_theta_sd = 10,
  normal_sigma_sd = 5,
  cauchy_scale = 2,
  iter_sampling = 4000,
  iter_warmup = 4000,
  chains = 4,
  thin_number = 1,
  adapt_delta = 0.95,
  max_treedepth = 15,
  AR_setting = 0,
  rho = 0,
  optimize_type = c("P", "G", "D"),
  L = 5e+05,
  draw_num = 10000
)
```

Arguments

Y	Numeric vector of responses of length `n`.
X	Design matrix of dimension `n x (p+1)`, including an intercept column. The second column must contain the raw covariate values.
V	Error covariance matrix of dimension `n x n`. Use `diag(n)` for i.i.d. errors (default).
a	Left endpoint of the covariate domain `[a, b]`.
b	Right endpoint of the covariate domain `[a, b]`.

theta_true	Numeric vector of true regression coefficients of length p+1. Used to evaluate ESCR in simulation studies. Set to NULL (default) when the true coefficients are unknown.
alpha	Nominal miscoverage rate. The credible band targets 1 - alpha simultaneous coverage. Default is 0.05.
prior_type	Character string specifying the prior on (theta, sigma). Either "normal_half_cauchy" (default, recommended) or "normal_normal".
normal_theta_sd	Prior standard deviation for each component of theta under the Normal-Normal prior. Default is 10.
normal_sigma_sd	Prior standard deviation for sigma under the Normal-Normal prior. Default is 5.
cauchy_scale	Scale parameter of the half-Cauchy prior on sigma under the Normal-half-Cauchy prior. Default is 2.
iter_sampling	Number of post-warmup HMC draws per chain. Default is 4000.
iter_warmup	Number of warmup draws per chain. Default is 4000.
chains	Number of Markov chains. Default is 4.
thin_number	Positive integer. Thinning interval for posterior draws. A value of k retains every k-th draw from each chain. Default is 1 (no thinning).
adapt_delta	Target acceptance probability for the NUTS sampler. Default is 0.95.
max_treedepth	Maximum tree depth for the NUTS sampler. Default is 15.
AR_setting	Integer. 0 for i.i.d. errors (default), 1 for AR(1) errors.
rho	AR(1) autocorrelation coefficient. Only used when AR_setting == 1. Default is 0.
optimize_type	Character. Method for computing $\sup_{x \in [a,b]} T(x)$: "P" = polyroot analytical method (recommended, default); "G" = global optimisation (grid-based); "D" = differential evolution (DEoptim).
L	Number of Monte Carlo draws used to estimate the critical constant lambda. Default is 500000.
draw_num	Number of Monte Carlo draws used to estimate the PSCP. Default is 10000.

Value

An object of class "bscb_fit", which is a list with the following components:

lambda	Estimated critical constant at level 1 - alpha.
lower_bound	Lower credible band evaluated on a fine grid over [a, b].
upper_bound	Upper credible band evaluated on a fine grid over [a, b].
theta_true	True regression coefficients (if supplied).
order_form	Polynomial order form used internally.
mu_star	Posterior mean of theta (length p+1).
cov_theta	Posterior covariance matrix of theta.

theta_mat	Matrix of posterior draws, (chains * floor(iter_sampling / thin_number)) x (p+1).
x_range	Numeric vector c(a, b).
call	The matched call.
method	Character string "HMC".
n	Sample size.
p	Polynomial degree.
alpha	Nominal miscoverage rate.
data	List containing the design matrix X and response vector Y.
lambda_samples	Numeric vector of length L containing the Monte Carlo supremum draws used to derive lambda.
params	List of additional settings: AR_setting, rho, prior_type, normal_theta_sd, normal_sigma_sd, cauchy_scale, iter_sampling, iter_warmup, chains, thin_number, L, draw_num, optimize_type.

See Also

compute_bscb_conjugate, compute_bscb_ind_jeffreys

Examples

```
set.seed(42)
n <- 20; p <- 2
x_seq <- seq(-5, 5, length.out = n)
X <- cbind(1, x_seq, x_seq^2)
theta_true <- c(-6, -3, 0.25)
Y <- as.numeric(X %*% theta_true + rnorm(n, sd = 0.2))
fit <- compute_bscb_hmc(
  Y = Y, X = X, V = diag(n),
  a = -5, b = 5,
  theta_true = theta_true,
  prior_type = "normal_half_cauchy",
  L = 1000, draw_num = 500   # small values for illustration only
)
fit$lambda
fit$params$prior_type
```

---

compute_bscb_ind_jeffreys

BSCB-J: Bayesian Simultaneous Credible Band under the Independent Jeffreys Prior

---

Description

Constructs a $(1 - \alpha)$ two-sided Bayesian simultaneous credible band for polynomial regression using the independent Jeffreys prior. The marginal posterior of θ follows a multivariate-t distribution with degrees of freedom $n - p - 1$.

Usage

```
compute_bscb_ind_jeffreys(
  X,
  Y,
  alpha = 0.05,
  a = NULL,
  b = NULL,
  L = 5e+05,
  AR_setting = 0,
  rho = NULL,
  optimize_type = c("P", "G", "D"),
  theta_true = NULL,
  verbose = TRUE
)
```

Arguments

`X`	Numeric matrix of dimension $n \times (p + 1)$. Design matrix with intercept in the first column.
`Y`	Numeric vector of length n. Response variable.
`alpha`	Numeric. Nominal mis-coverage level; the band targets $1 - \alpha$ simultaneous coverage. Default is `0.05`.
`a`	Numeric. Left endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
`b`	Numeric. Right endpoint of the covariate domain $[a, b]$. Inferred from `X[, 2]` if `NULL`.
`L`	Integer. Number of Monte Carlo draws for computing the critical constant λ. Default is `500000`.
`AR_setting`	Integer. Error covariance structure: `0` = i.i.d. errors (default); `1` = AR(1) errors.
`rho`	Numeric. AR(1) coefficient. Required when `AR_setting` = `1`.
`optimize_type`	Character. Method for computing $\sup_{x \in [a,b]} T(x)$: `"P"` = polyroot analytical method (recommended); `"G"` = global optimisation; `"D"` = differential evolution (DEoptim).
`theta_true`	Numeric vector of length $p + 1$. True regression coefficients. Optional; stored in the output for diagnostic use.
`verbose`	Logical. If `TRUE` (default), prints progress messages.

Value

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

lambda Critical constant for the credible band.

lower_bound Function: computes the lower band at a given x.

upper_bound Function: computes the upper band at a given x.

mu_star Posterior mean of θ (GLS estimate).

dof Degrees of freedom of the marginal posterior $(n - p - 1)$.

scale_mat Scale matrix Σ_0 of the marginal multivariate-t posterior distribution of θ.

cov_theta Posterior covariance matrix of θ. The posterior covariance matrix equals $\text{Cov}(\theta) = \frac{\nu}{\nu-2}\Sigma_0$, where ν is the degrees of freedom (dof).

x_range Covariate domain $[a, b]$.

lambda_samples Monte Carlo samples used to compute λ.

theta_true True parameters (if supplied).

method Character string "independent_jeffreys".

params List of configuration parameters.

Examples

```
# Quadratic model with i.i.d. errors
# This is for a quick demonstration;
# For actual use, please set L = 500,000.
set.seed(123)
n <- 50
x <- seq(-5, 5, length.out = n)
X <- cbind(1, x, x^2)
theta_true <- c(-6, -3, 0.25)
Y <- X %*% theta_true + rnorm(n, sd = 0.2)

fit <- compute_bscb_ind_jeffreys(
  X          = X,
  Y          = Y,
  alpha      = 0.05,
  a          = -5,
  b          =  5,
  L          = 5000,
  theta_true = theta_true,
  verbose    = FALSE
)

# Critical constant
fit$lambda

# Evaluate the band over a grid
x_seq     <- seq(-5, 5, length.out = 200)
lower_vec <- fit$lower_bound(x_seq)
upper_vec <- fit$upper_bound(x_seq)
```

```
# Full example with recommended L
fit_full <- compute_bscb_ind_jeffreys(
  X = X, Y = Y, alpha = 0.05, a = -5, b = 5,
  L = 50000, theta_true = theta_true
)
```

---

coverage_ESCR *Compute the coverage of BSCB*

Description

Compute the coverage of BSCB

Usage

```
coverage_ESCR(fit, optimize_type = c("P", "G", "D"), verbose = FALSE)
```

Arguments

fit	A BSCB fit object containing lambda, mu_star, cov_theta, theta_true, x_range, order_form
optimize_type	Character. Method for computing $\sup_{x\in[a,b]} T(x)$: "P" = polyroot function (recommended); "G" = global optimisation; "D" = Doptimize function from package DEoptim.
verbose	Logical. If TRUE (default), prints the value of the critical constant lambda.

Value

Integer: 1 if covered, 0 if not covered

Examples

```
# This is for a quick demonstration;
# For actual use, please set L = 500000.
set.seed(123)
n <- 50
p <- 2
x <- seq(-5, 5, length.out = n)
X <- cbind(1, x, x^2)
theta_true <- c(-6, -3, 0.25)

# Generate data and compute BSCB
Y <- X %*% theta_true + rnorm(n, 0, 0.2)
fit <- compute_bscb_conjugate(X, Y, alpha = 0.05, a = -5, b = 5,
                              L = 50000, theta_true = theta_true,
```

```
                                                verbose = FALSE)

# Check the empirical simultaneous coverage rate (ESCR)
is_covered <- coverage_ESCR(fit,  optimize_type ="P", verbose = TRUE)
cat("Coverage indicator:", is_covered, "\n")
```

---

coverage_PSCP *Compute the Posterior Simultaneous Coverage Probability (PSCP)*

Description

Estimates the posterior simultaneous coverage probability (PSCP) of a constructed BSCB by Monte Carlo integration over the posterior distribution of θ. For each posterior draw $\hat{\theta}$, the supremum $\sup_{x\in[a,b]} T(x)$ is computed and compared against the critical constant λ. The PSCP is the proportion of draws for which $\sup T(x) \leq \lambda$.

Usage

```
coverage_PSCP(
  fit,
  draw_num = 10000,
  optimize_type = c("P", "G", "D"),
  verbose = FALSE
)
```

Arguments

fit	An object of class "bscb_fit" returned by compute_bscb_conjugate or compute_bscb_ind_jeffreys Must contain lambda, mu_star, cov_theta, dof, x_range, and order_form.
draw_num	Integer. Number of Monte Carlo draws for estimating PSCP. Default is 10000.
optimize_type	Character. Method for computing $\sup_{x\in[a,b]} T(x)$: "P" = polyroot analytical method (recommended); "G" = global optimisation; "D" = differential evolution (DEoptim).
verbose	Logical. If TRUE, prints the estimated PSCP value. Default is FALSE.

Value

Numeric. Estimated posterior simultaneous coverage probability, a value in $[0, 1]$.

See Also

coverage_ESCR, compute_bscb_conjugate, compute_bscb_ind_jeffreys

Examples

```
# This is for a quick demonstration;
# For actual use, please set L = 500000 and draw_num = 10000.
set.seed(123)
n <- 50
x <- seq(-5, 5, length.out = n)
X <- cbind(1, x, x^2)
theta_true <- c(-6, -3, 0.25)
Y <- X %*% theta_true + rnorm(n, sd = 0.2)

fit <- compute_bscb_conjugate(
  X          = X,
  Y          = Y,
  alpha      = 0.05,
  a          = -5,
  b          =  5,
  L          = 1000,
  theta_true = theta_true,
  verbose    = FALSE
)

coverage_PSCP(fit, draw_num = 500, optimize_type = "P", verbose = TRUE)


# Full example with recommended draw_num
coverage_PSCP(fit, draw_num = 10000, optimize_type = "P")
```

---

create_order_form *Create a polynomial basis vector function*

Description

Returns a function that maps a scalar x to the polynomial basis vector $(1, x, x^2, \ldots, x^p)$.

Usage

```
create_order_form(p)
```

Arguments

p	Non-negative integer. Polynomial degree.

Value

A function `f(x)` that returns $(1, x, \ldots, x^p)$ as a numeric vector (for scalar `x`) or matrix (for vector `x`).

Examples

```
f <- create_order_form(p = 2)
f(3)   # returns c(1, 3, 9)
f(c(1, 2, 3))  # returns a matrix
```

`f_L_SCB` *Vertical distance from true curve to lower band boundary*

Description

Vertical distance from true curve to lower band boundary

Usage

```
f_L_SCB(x, cov_theta, mu_star, lambda_best_optim, theta_true)
```

Arguments

`x`	Numeric. Evaluation point.
`cov_theta`	Numeric matrix. Posterior covariance of theta.
`mu_star`	Numeric vector. Posterior mean of theta.
`lambda_best_optim`	Numeric. Critical constant lambda.
`theta_true`	Numeric vector. True regression coefficients.

Value

Numeric. Positive if true curve is above lower bound.

`f_U_SCB` *Vertical distance from true curve to upper band boundary*

Description

Vertical distance from true curve to upper band boundary

Usage

```
f_U_SCB(x, cov_theta, mu_star, lambda_best_optim, theta_true)
```

Arguments

x	Numeric. Evaluation point.
cov_theta	Numeric matrix. Posterior covariance of theta.
mu_star	Numeric vector. Posterior mean of theta.
lambda_best_optim	Numeric. Critical constant lambda.
theta_true	Numeric vector. True regression coefficients.

Value

Numeric. Positive if true curve is below upper bound.

---

find_global_maximum — *Find the global maximum of T(x) via grid search and local optimisation*

Description

Fallback method using a coarse grid search combined with `uniroot` and `optimize`. For most cases, `find_global_maximum_h_all` (Liu's analytic method) is preferred.

Usage

```
find_global_maximum(
  fn,
  a,
  b,
  order_form,
  theta,
  mu_star,
  cov_mat,
  tol = 1e-06,
  n_grid = 100
)
```

Arguments

fn	Function. The objective function T(x) to maximise.
a	Numeric. Left endpoint of the search interval.
b	Numeric. Right endpoint of the search interval.
order_form	Function. Polynomial basis function from `create_order_form`.
theta	Numeric vector. Posterior draw of theta.
mu_star	Numeric vector. Posterior mean of theta.
cov_mat	Numeric matrix. Covariance matrix.
tol	Numeric. Numerical tolerance. Default `1e-6`.
n_grid	Integer. Number of grid points. Default `100`.

Value

A list with components maximum, x_max, and all_candidates.

---

find_global_maximum_h_all

Find the global maximum of T(x) analytically via polyroot (Liu's method)

Description

Computes $\sup_{x \in [a,b]} T(x)$ by finding the stationary points of $h(x) = T(x)^2 = N(x)/D(x)$ via polyroot, where $N(x)$ and $D(x)$ are polynomials. This is the recommended method.

Usage

```
find_global_maximum_h_all(a, b, d, cov_mat)
```

Arguments

a	Numeric. Left endpoint of the search interval.
b	Numeric. Right endpoint of the search interval.
d	Numeric vector of length $p+1$. Direction vector ($\theta - \mu^*$).
cov_mat	Numeric matrix of dimension $(p+1) \times (p+1)$. Covariance matrix.

Value

A list with components maximum, x_max, and all_candidates.

---

fn_Bayes_ECR

T(x) for computing ESCR: uses true parameter theta_true

Description

T(x) for computing ESCR: uses true parameter theta_true

Usage

```
fn_Bayes_ECR(x, theta_true, mu_star, cov_theta)
```

Arguments

x	Numeric. Evaluation point.
theta_true	Numeric vector. True regression coefficients.
mu_star	Numeric vector. Posterior mean of theta.
cov_theta	Numeric matrix. Posterior covariance of theta.

Value

Numeric scalar. Value of T(x).

`fn_Bayes_PCP` *T(x) for computing lambda (PSCP): uses posterior draw theta_hat*

Description

T(x) for computing lambda (PSCP): uses posterior draw theta_hat

Usage

```
fn_Bayes_PCP(x, theta_hat, mu_star, cov_theta)
```

Arguments

`x`	Numeric. Evaluation point.
`theta_hat`	Numeric vector. Posterior draw of theta.
`mu_star`	Numeric vector. Posterior mean of theta.
`cov_theta`	Numeric matrix. Posterior covariance of theta.

Value

Numeric scalar. Value of T(x).

`fn_Freq_ECR` *T(x) to compute ESCR for frequentist methods*

Description

T(x) to compute ESCR for frequentist methods

Usage

```
fn_Freq_ECR(x, theta_true, lm_theta_hat, S, inv)
```

Arguments

`x`	Numeric. Evaluation point.
`theta_true`	Numeric vector. True regression coefficients.
`lm_theta_hat`	Numeric vector. OLS estimate of theta.
`S`	Numeric. Residual standard error.
`inv`	Numeric matrix. Inverse of $X^T X$.

Value

Numeric scalar. Value of T(x).

fn_neg_Bayes_ECR *T(x) for computing ESCR for DEoptim: uses true parameter theta_true*

Description

T(x) for computing ESCR for DEoptim: uses true parameter theta_true

Usage

```
fn_neg_Bayes_ECR(x, theta_true, mu_star, cov_theta)
```

Arguments

`x`	Numeric. Evaluation point.
`theta_true`	Numeric vector. True regression coefficients.
`mu_star`	Numeric vector. Posterior mean of theta.
`cov_theta`	Numeric matrix. Posterior covariance of theta.

Value

Numeric scalar. Value of T(x).

fn_neg_Bayes_PCP *Negative T(x) for minimisation-based optimisers (e.g. DEoptim)*

Description

Negative T(x) for minimisation-based optimisers (e.g. DEoptim)

Usage

```
fn_neg_Bayes_PCP(x, theta_hat, mu_star, cov_theta)
```

Arguments

`x`	Numeric. Evaluation point.
`theta_hat`	Numeric vector. Posterior draw of theta.
`mu_star`	Numeric vector. Posterior mean of theta.
`cov_theta`	Numeric matrix. Posterior covariance of theta.

Value

Numeric scalar. Negative value of T(x).

generate_simulation_data
Generate simulation datasets for polynomial regression

Description

Generates a fixed design matrix X and a list of response vectors Y for use in simulation studies of Bayesian simultaneous credible bands. The design can be either equally-spaced (ES) or D-optimal (DO).

Usage

```
generate_simulation_data(
  p,
  n,
  e_sd,
  theta_true,
  a = -5,
  b = 5,
  replication = 2,
  design_index = 2,
  center_index = 1,
  n_ES_x = n,
  n_DO_init_x = 3e+05,
  AR_index = 0,
  rho = 0.1,
  batch_index = 1,
  seed = NULL
)
```

Arguments

`p`	Integer. Polynomial degree. Must be 1, 2, or 3.
`n`	Integer. Sample size.
`e_sd`	Numeric. Error standard deviation (sigma in the paper).
`theta_true`	Numeric vector of length `p + 1`. True regression coefficients.
`a`	Numeric. Left endpoint of the covariate domain $[a, b]$.
`b`	Numeric. Right endpoint of the covariate domain $[a, b]$.
`replication`	Integer. Number of simulation replications.
`design_index`	Integer. Design type: `1` = equally-spaced (ES); `2` = D-optimal (DO).
`center_index`	Integer. Centering of covariates: `1` = mean-centred (default); `0` = uncentred; `2` = standardised.
`n_ES_x`	Integer. Number of equally-spaced design points. Only used when `design_index = 1`.

n_DO_init_x	Integer. Candidate pool size for D-optimal search. A large value (e.g. 300000) ensures that 6 support points are selected. Only used when design_index = 2.
AR_index	Integer. Error structure: 0 = i.i.d. (default); 1 = AR(1).
rho	Numeric. AR(1) coefficient. Only used when AR_index = 1.
batch_index	Integer. Batch index used as part of the random seed (set.seed(1000 * batch_index + i) for replication i).
seed	Integer or NULL. Base random seed for reproducibility for D-optimal design. If NULL (default), no seed is set and results will vary between runs.

Value

A list containing:

X Design matrix of dimension $n \times (p + 1)$.

Y.list List of replication response vectors, each of length n.

optimal_x Vector of selected support points.

optimal_weights Vector of observation counts at each support point.

Examples

```
# Example 1: quadratic model, D-optimal design
sim_data <- generate_simulation_data(
  p           = 2,
  n           = 20,
  e_sd        = 0.2,
  theta_true  = c(-6, -3, 0.25),
  a           = -5,
  b           =  5,
  replication = 1,
  design_index = 2,
  center_index = 1
)

X      <- sim_data$X
Y.list <- sim_data$Y.list



# Example 2: cubic model, equally-spaced design
sim_data2 <- generate_simulation_data(
  p            = 3,
  n            = 20,
  e_sd         = 0.2,
  theta_true   = c(1, 2, -1, 0.5),
  a            = -5,
  b            =  5,
  replication  = 1,
  design_index = 1,
  center_index = 1
```

```
)
```

L_SCB *Evaluate lower band at x*

Description

Evaluate lower band at x

Usage

```
L_SCB(x, cov_theta, mu_star, lambda_best_optim)
```

Arguments

`x`	Numeric. Evaluation point.
`cov_theta`	Numeric matrix. Posterior covariance of theta.
`mu_star`	Numeric vector. Posterior mean of theta.
`lambda_best_optim`	Numeric. Critical constant lambda.

Value

Numeric. Lower band value at x.

---

sup_T_Bayes_ESCR *To compute ESCR for BSCB and BPCB For BSCB use* `cov_mat = cov_theta`*; for BPCB use* `cov_mat = scale_mat`*.*

Description

To compute ESCR for BSCB and BPCB For BSCB use `cov_mat = cov_theta`; for BPCB use `cov_mat = scale_mat`.

Usage

```
sup_T_Bayes_ESCR(a, b, theta_true, mu_star, cov_mat)
```

Arguments

`a`	Numeric. Left endpoint.
`b`	Numeric. Right endpoint.
`theta_true`	Numeric vector. True regression coefficients.
`mu_star`	Numeric vector. Posterior mean of theta.
`cov_mat`	Numeric matrix. Covariance matrix.

Value

List with `maximum` and `x_max`.

`sup_T_Bayes_PSCP` *To compute the critical constant for BSCB and BPCB; To compute PSCP for BSCB and BPCB*

Description

To compute the critical constant for BSCB and BPCB; To compute PSCP for BSCB and BPCB

Usage

```
sup_T_Bayes_PSCP(a, b, theta_hat, mu_star, cov_mat)
```

Arguments

`a`	Numeric. Left endpoint.
`b`	Numeric. Right endpoint.
`theta_hat`	Numeric vector. Posterior draw of theta.
`mu_star`	Numeric vector. Posterior mean of theta.
`cov_mat`	Numeric matrix. Covariance matrix.

Value

List with `maximum` and `x_max`.

`sup_T_Freq_ESCR` *To compute the ESCR for FSCB and FPCB*

Description

To compute the ESCR for FSCB and FPCB

Usage

```
sup_T_Freq_ESCR(a, b, theta_true, lm_theta_hat, cov_mat)
```

Arguments

`a`	Numeric. Left endpoint.
`b`	Numeric. Right endpoint.
`theta_true`	Numeric vector. True regression coefficients.
`lm_theta_hat`	Numeric vector. OLS estimate of theta.
`cov_mat`	Numeric matrix. Scaled covariance matrix ($S^2 \times (X^T X)^{-1}$).

Value

List with maximum and x_max.

---

sup_T_simFSCB *To compute the critical constant for simFSCB*

Description

To compute the critical constant for simFSCB

Usage

```
sup_T_simFSCB(a, b, W_sample, cov_mat)
```

Arguments

a	Numeric. Left endpoint.
b	Numeric. Right endpoint.
W_sample	Numeric vector. Simulated draw.
cov_mat	Numeric matrix. Inverse of $X^T X$.

Value

List with maximum and x_max.

U_SCB *Evaluate upper band at x*

Description

Evaluate upper band at x

Usage

```
U_SCB(x, cov_theta, mu_star, lambda_best_optim)
```

Arguments

x	Numeric. Evaluation point.
cov_theta	Numeric matrix. Posterior covariance of theta.
mu_star	Numeric vector. Posterior mean of theta.
lambda_best_optim	Numeric. Critical constant lambda.

Value

Numeric. Upper band value at x.

Index