Package: fkbma (via r-universe)

January 31, 2025

Type Package

Title Free Knot-Bayesian Model Averaging

Version 0.1.0

Maintainer Lara Maleyeff <lara.maleyeff@mcgill.ca>

Description Analysis of Bayesian adaptive enrichment clinical trial using Free-Knot Bayesian Model Averaging (FK-BMA) method of Maleyeff et al. (2024) for Gaussian data. Maleyeff, L., Golchi, S., Moodie, E. E. M., & Hudson, M. (2024) ``An adaptive enrichment design using Bayesian model averaging for selection and threshold-identification of predictive variables'' <doi:10.1093/biomtc/ujae141>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports rstan, matrixStats, splines, dplyr, coda, MASS, ggplot2, cowplot, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

NeedsCompilation no

Author Lara Maleyeff [aut, cre], Shirin Golchi [aut], Erica E. M. Moodie [aut]

Repository CRAN

Date/Publication 2025-01-31 10:50:02 UTC

Config/pak/sysreqs make

Contents

coef.rjMCMC	*Extract Posterior Mean Coefficients from rjMCMC Results*

Description

This function extracts the posterior means of the intercept, treatment parameters, and binary parameters from the results of an rjMCMC model.

Usage

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

Arguments

object	An object of class rjMCMC containing the output from the `rjMCMC` procedure, including posterior samples.
...	Additional arguments to be passed to other methods or functions.

Value

A numeric vector containing the posterior mean of the intercept, treatment, and binary parameters.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
coef(results)
```

---

credint	*Credible Intervals for rjMCMC Results*

Description

This function calculates the posterior mean and credible intervals for parameters from the rjMCMC results, including both intercept/treatment parameters and binary parameters. The credible intervals are computed based on the specified confidence level.

Usage

```
credint(results, level = 0.95)
```

Arguments

results	An object of class rjMCMC containing the output from the rjMCMC procedure, including posterior samples.
level	The level for the credible intervals (default is 0.95).

Value

A data frame with estimates, lower, and upper bounds of the credible intervals.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)
```

```
mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
credint(results)
```

---

fitted.rjMCMC   *Fitted values from Reversible Jump MCMC (rjMCMC) Model Results*

Description

This function generates posterior fitted values from an rjMCMC model based on the provided data. It combines the fixed effects, spline terms, and binary parameters.

Usage

```
## S3 method for class 'rjMCMC'
fitted(object, newdata = NULL, ...)
```

Arguments

object	An object of class rjMCMC containing the output from the rjMCMC procedure, including posterior samples and fitted splines.
newdata	A data frame for which fited values are to be computed. If NA, the original fitted data is used.
...	Additional arguments to be passed to other methods or functions.

Value

A matrix of fitted values.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
```

```
newdata = data.frame(Z_1 = 1, Z_2 = 1, Z_3 = 1, Z_4 = 1, Z_5 = 1,
                     trt = 1, X_1 = seq(0,1,by=0.01))
fitted(results)
fitted(results,newdata)
```

---

fittedTrtEff	*Fitted treatment effect values from Reversible Jump MCMC (rjMCMC)* *Model Results*

Description

This function generates posterior fitted treatment effects from an rjMCMC model based on the provided data. It combines the fixed effects, spline terms, and binary parameters.

Usage

```
fittedTrtEff(results, newdata = NULL)
```

Arguments

results   An object of class rjMCMC containing the output from the `rjMCMC` procedure, including posterior samples and fitted splines.

newdata   A data frame for which fitted values are to be computed. If NA, the original fitted data is used.

Value

A matrix of fitted values.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
newdata = data.frame(Z_1 = 1, Z_2 = 1, Z_3 = 1, Z_4 = 1, Z_5 = 1,
                     trt = 1, X_1 = seq(0,1,by=0.01))
fittedTrtEff(results)
fittedTrtEff(results,newdata)
```

`getEffectiveSubspace`   *Get Effective Subspace*

Description

This function identifies the "effective subspace" where treatment is effective based on posterior inference results from the FK-BMA model. It analyzes interaction terms between treatment and covariates, allowing for both binary and continuous variables.

Usage

```
getEffectiveSubspace(results, newdata = NULL, alpha = 0.05, pip_cutoff = 0.1)
```

Arguments

results	A fitted model object from `rjMCMC`.
newdata	Optional. A new dataset for evaluating the effective subspace. If `NULL`, the function uses `results$data_fit`.
alpha	Numeric. The alpha level used for computing quantiles. Default is `0.05`.
pip_cutoff	Numeric. The minimum Posterior Inclusion Probability (PIP) threshold for selecting covariates. Default is `0.1`.

Details

- The function computes the posterior treatment effect for each observation in the dataset using the `fittedTrtEff` function and evaluates its quantiles at the specified `alpha` level.

- Binary variables with high posterior inclusion probabilities (PIP) are used to define subgroups, and the corresponding effective subspaces for a continuous variable are identified by checking where the treatment effect quantiles are strictly positive.

- If the number of binary variables is <= 3 and there is exactly one continuous variable, the function describes the effective subspace in terms of disjoint intervals.

- For more complex cases, a warning is issued suggesting alternative methods such as Bayesian regression trees for interpretation.

Value

A list with the following components:

quantiles	A vector of quantile values for the treatment effect in the new dataset.
is_effective_subspace	
	A logical vector indicating whether the treatment effect is positive in the effective subspace.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
getEffectiveSubspace(results)
```

---

pip   *Compute Posterior Inclusion Probabilities (PIPs) for rjMCMC Results*

Description

This function returns the posterior inclusion probabilities (PIPs) for all variables, including the intercept and treatment, based on the results from an rjMCMC model.

Usage

```
pip(results)
```

Arguments

results   An object of class rjMCMC containing the output from the rjMCMC procedure, including the PIP summary for variables.

Value

A numeric vector with the PIPs for the intercept, treatment, and other variables.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
```

```
                        degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
pip(results)
```

---

plot.rjMCMC   *Plotting function for* rjMCMC *results*

Description

This function generates plots for model results from rjMCMC based on specified sample type, effect type, and plot type. The function is flexible for various combinations of sample_type, effect_type, and plot_type, as outlined below.

Usage

```
## S3 method for class 'rjMCMC'
plot(
  x,
  ...,
  variables = NULL,
  sample_type = "fitted",
  effect_type = "treatment_effect",
  plot_type = "cred",
  level = 0.95,
  aux_vars = list(),
  facet_by = NULL,
  pip_cutoff = 0.1
)
```

Arguments

x	A fitted model object from rjMCMC.
...	Additional arguments to be passed to other methods or functions.
variables	A vector of variable names to include in the plot. Automatically set to continuous variables if NULL.
sample_type	Character string specifying the type of sample: "fitted", "predictive", or "estimand". "fitted" and "predictive" are compatible with plot_type = "cred". "estimand" is compatible with plot_type = "hist" or "trace" (only used for individual parameter trajectories).
effect_type	Character string indicating the effect type: "treatment_effect" or "outcome". For "treatment_effect", the function plots the fitted or predictive effect of treatment; for "outcome", it plots outcome values without interaction with treatment.

plot_type	Character string specifying the plot type: "cred" for credible interval plots, or "hist"/"trace" for histogram or trace plots of individual parameters (only for sample_type = "estimand").
level	Numeric value for the credible interval level (default is 0.95).
aux_vars	A list of auxiliary variables and their fixed values. Each element name must match a model variable.
facet_by	A vector of variable names to facet by in the plot. Automatically set to binary model variables if NULL.
pip_cutoff	Numeric threshold for the posterior inclusion probability (PIP) of model variables to include in the plot.

Details

- **Sample and Plot Compatibility**:
  - For sample_type = "estimand", only plot_type = "hist" or "trace" is allowed, as these are designed to visualize the posterior distribution or MCMC trajectory of individual parameters. Parameters like intercept, trt, and sigma are agnostic to effect_type as they do not interact with treatment.
  - plot_type = "cred" is designed for use with sample_type = "fitted" or "predictive" and shows credible intervals for the outcome (y-axis) across biomarker values (x-axis) by covariate pattern. effect_type controls whether the treatment effect or main effect is displayed.

- **Effect Types**:
  - outcome plots either the fitted or predictive values without treatment interaction, allowing for treatment (trt) values to be specified.
  - treatment_effect plots the interaction of the treatment effect across different covariate patterns.

Value

A ggplot2 object or a grid of plots.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
plot(results, sample_type = "fitted", effect_type = "treatment_effect", plot_type = "cred")
```

```
plot(results, sample_type = "estimand", plot_type = "hist")
```

---

predict.rjMCMC   *Predict from Reversible Jump MCMC (rjMCMC) Model Results*

Description

This function generates posterior predictions from an rjMCMC model based on the provided data.
It combines the fixed effects, spline terms, and binary parameters, and includes residual variance in
the predictions.

Usage

```
## S3 method for class 'rjMCMC'
predict(object, newdata = NULL, ...)
```

Arguments

object   An object of class rjMCMC containing the output from the rjMCMC procedure,
                including posterior samples and fitted splines.

newdata   A data frame for which predictions are to be made. If NA, the original fitted
                data is used.

...   Additional arguments to be passed to other methods or functions.

Value

A matrix of predicted values.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
newdata = data.frame(Z_1 = 1, Z_2 = 1, Z_3 = 1, Z_4 = 1, Z_5 = 1,
                     trt = 1, X_1 = seq(0,1,by=0.01))
predict(results)
predict(results,newdata)
```

---

predictTrtEff   *Predict Treatment Effect*

Description

This function predicts the treatment effect for new data based the Reversible Jump MCMC (rjM-CMC) results.

Usage

```
predictTrtEff(results, newdata = NULL)
```

Arguments

results	An object of class rjMCMC containing the output from the `rjMCMC` procedure, including posterior samples and fitted splines.
newdata	A data frame for which predicted values are to be computed If NA, the original fitted data is used.

Details

The function:

- Checks if the required columns in `results$candinter` are present in `newdata`.
- Computes the fitted posterior treatment effect based on main treatment effects, spline interactions, and binary interactions.
- Adds noise to the fitted posterior using the residual variance `results$sigma_sq` to generate predictive posterior samples.

Spline interactions are handled by interpolating the spline coefficients for the values in `newdata`.

Value

A matrix of predictive posterior samples for the treatment effect, where each row corresponds to a posterior sample and each column corresponds to an observation in `newdata`.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))
```

```
results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
newdata = data.frame(Z_1 = 1, Z_2 = 1, Z_3 = 1, Z_4 = 1, Z_5 = 1,
                     trt = 1, X_1 = seq(0,1,by=0.01))
predictTrtEff(results)
predictTrtEff(results,newdata)
```

---

print.rjMCMC	*Print a summary of results from from Reversible Jump MCMC (rjM-CMC)*

Description

This function provides a detailed summary of the results from the `rjMCMC` procedure, including model information, parameter estimates, posterior inclusion probabilities, convergence diagnostics, and plots for spline terms. The function also prints the model formula with `fbs()` notation for spline terms, indicating the use of free-knot B-splines.

Usage

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

Arguments

x   An object of class rjMCMC containing the output from the `rjMCMC` procedure, which includes:

   success Logical indicating whether the sampler converged based on Geweke diagnostics.

   inter_trt_param Matrix of posterior samples for treatment intercept and main effect.

   binary_param Matrix of posterior samples for binary variable parameters.

   sigma_sq Matrix of posterior samples for the residual variance (sigma squared).

   vars_prop_summ Posterior inclusion probabilities for candidate variables.

   splines_fitted List of matrices containing fitted values for spline terms across iterations.

   data_fit Original dataset used in the `rjMCMC` procedure.

   candsplineinter Names of continuous candidate predictive spline variables.

   candsplinevars Names of continuous candidate spline variables.

   candbinaryvars Names of binary candidate variables.

   candinter Names of interaction terms, which can include spline variables.

   mcmc_specs MCMC sampler specifications, including the number of samples, burn-in, thinning, and chains.

...   Additional arguments to be passed to other methods or functions.

Details

The function produces detailed summaries similar to those from `brms`, including diagnostics, estimates, posterior inclusion probabilities, and spline effects. The spline terms are wrapped in `fbs()` notation, indicating the use of free-knot B-splines in the model. If the sampler did not converge, a warning is issued. The function also allows the user to view diagnostic plots for fitted treatment effects.

Value

Prints the following summary information:

Model Formula The model formula with spline terms wrapped in `fbs()`, indicating free-knot B-splines, and interaction terms appropriately formatted.

Convergence Diagnostics Reports any convergence issues based on Geweke diagnostics.

MCMC Sampler Arguments Displays MCMC sampler arguments, including the number of posterior samples, burn-in, thinning, and chains.

Parameter Estimates Posterior mean, standard error, 95% credible intervals, effective sample size (ESS), Gelman-Rubin statistic (Rhat), and posterior inclusion probabilities (PIP) for binary parameters, treatment intercept, and treatment effect.

Gaussian Family Parameters Posterior summary for the residual standard error (sigma).

Posterior Inclusion Probabilities for Splines Prints the posterior inclusion probabilities for spline terms.

Plots for Fitted Treatment Effects Plots the mean and 95% credible intervals for each spline term vs fitted treatment effects.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
print(results)
```

---

rhats   *Calculate and Print Rhat Diagnostics for rjMCMC Results*

Description

This function calculates the Rhat diagnostic for convergence based on the posterior samples of individual treatment effects, intercept, and main treatment effect from an rjMCMC model. It prints the median, minimum, and maximum Rhat values for the treatment effects, as well as the Rhat for the intercept and treatment effect.

Usage

```
rhats(results)
```

Arguments

results   An `rjMCMC` object, including:

    trt_eff_posterior Matrix of posterior treatment effects.

    inter_trt_param Matrix of posterior estimates for intercept and main treatment
        effect.

Details

This function calculates R-hat statistics to assess MCMC convergence for both treatment effects and model parameters. Diagnostic plots are generated to visually inspect the chains across iterations.

Value

A list containing:

Rhat_trt_eff_posterior Vector of R-hat values for each individual's treatment effect.

Rhat_inter R-hat value for the intercept parameter.

Rhat_trt R-hat value for the main effect of treatment.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                  degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
```

```
                        mcmc_specs, prior_params)
    rhats(results)
```

---

rjMCMC   *Run Reversible Jump MCMC (rjMCMC) Procedure*

Description

This function performs a Reversible Jump MCMC (rjMCMC) procedure to generate the posterior distribution, using Bayesian model averaging and free-knot B-splines.

Usage

```
rjMCMC(
  data,
  candsplinevars,
  candbinaryvars,
  candinter,
  mcmc_specs = NULL,
  prior_params = NULL
)
```

Arguments

data	A data frame containing the observations, including the following columns:
	trt Group indicator (=1 for experimental group; =0 for control group).
	Y Continuous-valued outcome.
	candsplinevars All candidate spline variables as described in candsplinevars.
	candbinaryvars All candidate binary variables as described in candbinaryvars.
candsplinevars	A vector of names for continuous predictive candidate variables (default = NULL).
candbinaryvars	A vector of names for binary predictive candidate variables (default = NULL).
candinter	A vector indicating which of the candidate variables are tailoring (default = NULL).
mcmc_specs	A list containing:
	B Number of posterior samples (default = 2000).
	burnin Number of burn-in samples (default = 10000).
	thin Thinning parameter (default = 5).
	chain Number of chains (default = 1).
	sigma_v Proposal variance for "jump" terms (default = 0.1).
	bma Boolean indicating whether to include Bayesian model averaging step (default = TRUE).
prior_params	A list containing prior parameters:
	lambda_1 Prior parameter for the number of terms in the model (default = 0.1).

lambda_2 Prior parameter for the number of knots in each spline (default = 1).

a_0 Shape parameter for inverse gamma prior on individual-level variance (default = 0.01).

b_0 Rate parameter for inverse gamma prior on individual-level variance (default = 0.01).

degree Degree of B-splines (default = 3).

k_max Maximum number of knots for each spline term (default = 9).

w Window for proposing knot location changes (default = 1).

sigma_B Prior normal variance for model coefficients (default = sqrt(20)).

Value

An rjMCMC object with the following components:

success Logical indicating whether the MCMC chains converged based on Geweke diagnostics.

geweke.trt_eff_posterior Geweke diagnostic for the posterior treatment effects across the dataset.

geweke.sd Geweke diagnostic for the posterior distribution of residual variance (sigma squared).

accept_var Matrix indicating the acceptance of variable inclusion/removal for each iteration.

accept_add_knot Matrix indicating acceptance of knot addition for each spline term across iterations.

accept_remove_knot Matrix indicating acceptance of knot removal for each spline term across iterations.

accept_move_knot Matrix indicating acceptance of knot movement for each spline term across iterations.

splines_fitted List of matrices, one per spline interaction term, containing fitted spline values across iterations.

binary_param Matrix containing posterior samples of binary variable parameters.

inter_trt_param Matrix containing posterior samples of the treatment intercept and main effect.

sigma_sq Matrix of posterior samples for the residual variance (sigma squared).

vars_prop Matrix indicating the inclusion of variables across iterations (1 for included, 0 for excluded).

vars_prop_summ Posterior inclusion probabilities for all candidate variables (spline and binary).

k Matrix indicating the number of knots for each spline term across iterations.

trt_eff_posterior Matrix of posterior treatment effect estimates, including spline effects.

data_fit The original dataset passed to the function.

candsplineinter A character vector indicating the spline interaction terms.

candsplinevars A character vector of candidate spline variables.

candbinaryvars A character vector of candidate binary variables.

candinter A character vector of interaction terms with treatment (can include splines and binary variables).

mcmc_specs The MCMC specifications used in the procedure.

prior_params The prior parameters used in the procedure.

References

Maleyeff, L., Golchi, S., Moodie, E. E. M., & Hudson, M. (2024) "An adaptive enrichment design using Bayesian model averaging for selection and threshold-identification of predictive variables" doi:10.1093/biomtc/ujae141

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
```

---

simulated_data	*A synthetic dataset with continuous and binary covariates and a binary treatment variable (*`trt`*).*

Description

This dataset was generated using the formula: $Y = 2*Z_1 + 2*X_1 + 2*Z_1*trt + \cos(X_1 \cdot 2\pi)*trt + \epsilon$, where $\epsilon \sim N(0, 0.1)$.

Usage

```
data(simulated_data)
```

Format

A data frame with 1000 rows and 8 variables:

X_1 A continuous variable sampled from `U(0, 1)`.

Z_1 A binary variable sampled from `Bernoulli(0.35)`.

Z_2 A binary variable sampled from `Bernoulli(0.5)`.

Z_3 A binary variable sampled from `Bernoulli(0.65)`.

Z_4 A binary variable sampled from `Bernoulli(0.2)`.

Z_5 A binary variable sampled from `Bernoulli(0.35)`.

trt A binary treatment variable sampled from `Bernoulli(0.5)`.

Y The outcome variable calculated using the formula above.

Examples

```
data(simulated_data)
head(simulated_data)
```

---

summary.rjMCMC   *Summarize Results from Reversible Jump MCMC (rjMCMC)*

Description

This function provides a detailed summary of the results from the `rjMCMC` procedure, including model information, parameter estimates, posterior inclusion probabilities, convergence diagnostics, and plots for spline terms. The function also prints the model formula with `fbs()` notation for spline terms, indicating the use of free-knot B-splines.

Usage

```
## S3 method for class 'rjMCMC'
summary(object, digits = 2, level = 0.95, ...)
```

Arguments

object	An object of class rjMCMC containing the output from the `rjMCMC` procedure, which includes:

    success Logical indicating whether the sampler converged based on Geweke diagnostics.

    inter_trt_param Matrix of posterior samples for treatment intercept and main effect.

    binary_param Matrix of posterior samples for binary variable parameters.

    sigma_sq Matrix of posterior samples for the residual variance (sigma squared).

    vars_prop_summ Posterior inclusion probabilities for candidate variables.

    splines_fitted List of matrices containing fitted values for spline terms across iterations.

    data_fit Original dataset used in the `rjMCMC` procedure.

    candsplineinter Names of continuous candidate predictive spline variables.

    candsplinevars Names of continuous candidate spline variables.

    candbinaryvars Names of binary candidate variables.

    candinter Names of interaction terms, which can include spline variables.

    mcmc_specs MCMC sampler specifications, including the number of samples, burn-in, thinning, and chains.

digits	Number of digits in summary output
level	Credible interval level
...	Additional arguments to be passed to other methods or functions.

Details

The function produces detailed summaries similar to those from `brms`, including diagnostics, estimates, posterior inclusion probabilities, and spline effects. The spline terms are wrapped in `fbs()` notation, indicating the use of free-knot B-splines in the model. If the sampler did not converge, a warning is issued. The function also allows the user to view diagnostic plots for fitted treatment effects.

Value

Prints the following summary information:

Model Formula The model formula with spline terms wrapped in `fbs()`, indicating free-knot B-splines, and interaction terms appropriately formatted.

Convergence Diagnostics Reports any convergence issues based on Geweke diagnostics.

MCMC Sampler Arguments Displays MCMC sampler arguments, including the number of posterior samples, burn-in, thinning, and chains.

Parameter Estimates Posterior mean, standard error, 95% credible intervals, effective sample size (ESS), Gelman-Rubin statistic (Rhat), and posterior inclusion probabilities (PIP) for binary parameters, treatment intercept, and treatment effect.

Gaussian Family Parameters Posterior summary for the residual standard error (sigma).

Posterior Inclusion Probabilities for Splines Prints the posterior inclusion probabilities for spline terms.

Plots for Fitted Treatment Effects Plots the mean and 95% credible intervals for each spline term vs fitted treatment effects.

Examples

```
# Example dataset
data("simulated_data")

candsplinevars <- c("X_1")
candbinaryvars <- paste0("Z_", 1:5)
candinter <- c(candsplinevars, candbinaryvars)

mcmc_specs <- list(B = 2000, burnin = 1000, thin = 1, chains = 2, sigma_v = 0.1, bma = TRUE)
prior_params <- list(lambda_1 = 0.1, lambda_2 = 1, a_0 = 0.01, b_0 = 0.01,
                     degree = 3, k_max = 9, w = 1, sigma_B = sqrt(20))

results <- rjMCMC(simulated_data, candsplinevars, candbinaryvars, candinter,
                  mcmc_specs, prior_params)
summary(results)
```

Index