Package 'pHMC'

August 21, 2026

Type Package

Title Proximal Hamiltonian Monte Carlo for Non-Smooth Bayesian Inference

Version 0.1.0

Description Implements the Proximal Hamiltonian Monte Carlo (p-HMC) algorithm for Bayesian sampling and estimation from non-differentiable target densities. The method decomposes a target potential into a smooth component f(x) and a non-smooth convex component g(x), approximating only g(x) via its Moreau-Yosida envelope while retaining exact gradient information for f(x). This approach, based on the methodology described in Shukla, Vats, and Chi (2025) <doi:10.48550/arXiv.2510.22252>, yields improved Hamiltonian conservation over full-potential smoothing approaches. The package provides generalized routines accepting user-defined probability density functions, log-likelihoods, priors, and proximal operators, together with automated hyperparameter tuning for the Moreau-Yosida regularization parameter, Markov chain Monte Carlo convergence diagnostics, effective sample size computation, and model evaluation metrics including the Akaike information criterion and Bayesian information criterion.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, graphics, grDevices, utils, Matrix

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID: <https://orcid.org/0000-0003-1606-0844>), Arvind Pandey [aut], Bhupendra Singh [aut], Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-21 12:40:09 UTC

Contents

---

phmc *Generalized Proximal Hamiltonian Monte Carlo Sampler and Estimator*

Description

Fits Bayesian models and estimates parameters using the Proximal Hamiltonian Monte Carlo (p-HMC) algorithm for non-differentiable target densities as proposed by Shukla, Vats, and Chi (2025).

Usage

```
phmc(
  fn,
  grad_f = NULL,
  g = NULL,
  prox_fn = "l1",
  start,
  data = NULL,
  n_draws = 2000,
  burnin = floor(n_draws/2),
  thin = 1,
  epsilon = 0.01,
  L = 10,
  lambda_g = 0.01,
  M = NULL,
  tune_lambda = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

fn	Function. The smooth potential component $f(x)$ or complete potential $U(x) = -\log \pi(x)$. Must accept parameter vector x as first argument.
grad_f	Function or NULL. Gradient of the smooth potential component $f(x)$. If NULL, finite difference approximation is automatically computed.
g	Function or NULL. The non-smooth penalty component $g(x)$.
prox_fn	Function or character. Proximal mapping operator for $g(x)$, or name of built-in operator ("l1", "l2", "elastic_net", "nuclear_norm", "none"). Default is "l1".
start	Numeric vector or matrix. Initial parameter values.
data	Optional dataset passed as second argument to fn, grad_f, and g.
n_draws	Integer > 0. Total number of MCMC iterations to run (default 2000).
burnin	Integer >= 0. Number of initial draws to discard as burn-in (default floor(n_draws / 2)).
thin	Integer >= 1. Thinning interval (default 1).
epsilon	Numeric scalar > 0. Leapfrog step size (default 0.01).
L	Integer >= 1. Number of leapfrog steps per proposal (default 10).
lambda_g	Numeric scalar > 0. Moreau-Yosida regularization parameter (default 0.01).
M	Mass matrix or NULL (defaults to identity matrix).
tune_lambda	Logical. If TRUE, uses phmc_tune to automatically select optimal lambda_g.
verbose	Logical. If TRUE, prints sampling progress.
...	Additional arguments passed to fn, grad_f, g, and prox_fn.

Value

An object of class "phmc", which is a list containing:

draws	A numeric matrix of class "matrix" containing MCMC parameter samples after burn-in and thinning.
estimates	A summary matrix of class "matrix" with rows corresponding to parameters and columns containing posterior Mean, MAP, Median, StdErr, 2.5% and 97.5% credible interval bounds, ESS, and ESS per second.
accept_rate	A numeric scalar of class "numeric" giving the overall Metropolis-Hastings acceptance rate (between 0 and 1).
ess	A named numeric vector of class "numeric" containing Effective Sample Size estimates for each parameter.
ess_per_sec	A named numeric vector of class "numeric" containing Effective Sample Size per second for each parameter.
logLik	A numeric scalar of class "numeric" giving the log-likelihood value evaluated at the Maximum A Posteriori (MAP) parameter estimate.
AIC	A numeric scalar of class "numeric" giving the Akaike Information Criterion value for model assessment.

BIC	A numeric scalar of class "numeric" giving the Bayesian Information Criterion value for model assessment.
DIC	A numeric scalar of class "numeric" giving the Deviance Information Criterion value for model assessment.
elapsed_time	A numeric scalar of class "numeric" giving total sampler execution time in seconds.
lambda_g	A numeric scalar of class "numeric" specifying the Moreau-Yosida regularization parameter used during sampling.
call	An object of class "call" recording the matched function call.

References

Shukla A, Vats D, Chi EC (2025). "Proximal Hamiltonian Monte Carlo." *arXiv preprint*, doi:10.48550/arXiv.2510.22252.

Examples

```
set.seed(42)
y_data <- rnorm(100, mean = 1, sd = 0.5)
f_smooth <- function(x, y) 0.5 * sum((y - x)^2)
grad_f_smooth <- function(x, y) -sum(y - x)
fit <- phmc(fn = f_smooth, grad_f = grad_f_smooth,
            prox_fn = "l1", start = 0.5,
            data = y_data, lambda_g = 0.01,
            n_draws = 500)
summary(fit)
```

---

phmc_methods *S3 Methods for Proximal Hamiltonian Monte Carlo Objects*

Description

Provides summary, printing, extraction, and diagnostic plotting methods for objects returned by phmc.

Usage

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

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

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

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

## S3 method for class 'phmc'
vcov(object, ...)

## S3 method for class 'phmc'
logLik(object, ...)

## S3 method for class 'phmc'
plot(x, type = c("all", "trace", "acf", "density"), par_indices = NULL, ...)
```

Arguments

x	An object of class "phmc".
...	Additional arguments passed to generic methods.
object	An object of class "phmc".
type	Character string specifying plot type: "trace" for trace plots, "acf" for autocorrelation, "density" for posterior density, or "all" (default).
par_indices	Optional integer vector specifying parameter indices to plot.

Value

Depending on the S3 method invoked, returns the following:

`print.phmc` Invisibly returns the input object x of class "phmc" (called for its side effect of printing summary metrics to the console).

`summary.phmc` Returns an object of class "summary.phmc", which is a list containing model parameter estimates, acceptance rate, log-likelihood, information criteria (AIC, BIC, DIC), elapsed time, and regularization parameter lambda_g.

`print.summary.phmc` Invisibly returns the input object x of class "summary.phmc" (called for its side effect of printing detailed summary results to the console).

`coef.phmc` Returns a named numeric vector of class "numeric" containing posterior mean parameter estimates.

`vcov.phmc` Returns a numeric matrix of class "matrix" containing the empirical posterior variance-covariance matrix of the MCMC parameter draws.

`logLik.phmc` Returns an object of class "logLik" representing the log-likelihood value evaluated at the MAP estimate, with attributes "df" (number of estimated parameters) and "nobs" (number of retained MCMC draws).

`plot.phmc` Invisibly returns the input object x of class "phmc" (called for its side effect of generating diagnostic MCMC trace plots, autocorrelation functions, and posterior density curves).

phmc_tune *Hyperparameter Tuning for Moreau-Yosida Regularization Parameter*

Description

Evaluates the relative Hamiltonian error metric across a grid of candidate `lambda_g` values to select the optimal regularization parameter balancing potential smoothness and Hamiltonian conservation, as detailed in Section V of Shukla, Vats, and Chi (2025).

Usage

```
phmc_tune(
  fn,
  grad_f = NULL,
  g = NULL,
  prox_fn = "l1",
  start,
  lambda_grid = 10^seq(-5, 0, length.out = 15),
  epsilon = 0.001,
  L = 10,
  M = NULL,
  target_rel_err = 1e-04,
  data = NULL,
  seed = NULL,
  ...
)
```

Arguments

`fn`	Function. Smooth component $f(x)$ of the potential. Must accept parameter vector as first argument.
`grad_f`	Function or `NULL`. Analytical gradient of $f(x)$. If `NULL`, finite differences are used.
`g`	Function or `NULL`. Non-smooth component $g(x)$.
`prox_fn`	Function or character. Proximal operator or built-in name (`"l1"`, `"l2"`, `"nuclear_norm"`, `"elastic_net"`, `"none"`).
`start`	Numeric vector. Starting parameter value.
`lambda_grid`	Numeric vector. Grid of candidate `lambda_g` values to test.
`epsilon`	Numeric scalar > 0. Leapfrog step size.
`L`	Integer >= 1. Number of leapfrog steps.
`M`	Mass matrix or `NULL` (defaults to identity matrix).
`target_rel_err`	Numeric scalar. Maximum acceptable relative Hamiltonian error (default 1e-4).
`data`	Optional dataset passed as second argument to `fn` and `grad_f`.
`seed`	Optional integer or `NULL`. Random seed to set conditionally for reproducible initial momentum sampling. Default is `NULL`.
`...`	Additional arguments forwarded to `fn`, `grad_f`, or `prox_fn`.

Details

The relative Hamiltonian error metric is defined as

$$R_{\lambda_g} = \left| \frac{H(x_0, p_0) - H(\tilde{T}^{\lambda_g}_{\epsilon,L}(x_0, p_0))}{H(x_0, p_0)} \right|.$$

Value

An object of class `"phmc_tune"`, which is a list containing:

`optimal_lambda_g`
: A numeric scalar of class `"numeric"` specifying the selected optimal Moreau-Yosida regularization parameter `lambda_g` that satisfies the relative Hamiltonian error threshold.

`grid_results`
: A data frame of class `"data.frame"` containing tested candidate `lambda_g` values, computed relative Hamiltonian errors (`R_lambda_g`), and absolute Hamiltonian difference metrics (`H_diff`).

`target_rel_err`
: A numeric scalar of class `"numeric"` giving the target relative Hamiltonian error threshold used during grid search tuning.

References

Shukla A, Vats D, Chi EC (2025). "Proximal Hamiltonian Monte Carlo." *arXiv preprint*, doi:10.48550/arXiv.2510.22252.

`proximal_operators` *Proximal Mapping Operators and Moreau-Yosida Envelope Gradients*

Description

Evaluates proximal operators for common non-smooth penalties (L1 norm, nuclear norm, L2 norm, elastic net) and computes the gradient of the Moreau-Yosida envelope as described in Shukla, Vats, and Chi (2025).

Usage

```
prox_l1(x, tau)

prox_l2(x, tau)

prox_elastic_net(x, tau, alpha = 0.5)

prox_nuclear(x, tau)

grad_my_envelope(x, prox_fn = "l1", lambda_g = 0.01, ...)
```

Arguments

x	Numeric vector or matrix. Parameter value at which to evaluate the operator.
tau	Numeric scalar. Thresholding parameter (typically lambda_g * scale).
alpha	Numeric scalar in [0, 1]. Mixing parameter for elastic net penalty.
prox_fn	Function or character string. Proximal mapping function or name of built-in proximal operator ("l1", "l2", "elastic_net", "nuclear_norm", "none").
lambda_g	Numeric scalar > 0. Moreau-Yosida regularization parameter.
...	Additional arguments passed to the proximal operator.

Details

The Moreau-Yosida envelope of a proper, lower-semicontinuous, convex function g with scaling parameter $\lambda_g > 0$ is defined as

$$g_{\lambda_g}(x) = \inf_y \{g(y) + \frac{1}{2\lambda_g}\|y - x\|^2\}.$$

Its gradient is given by

$$\nabla g_{\lambda_g}(x) = \frac{1}{\lambda_g}(x - \text{prox}^g_{\lambda_g}(x)).$$

Value

For prox_l1, prox_l2, prox_elastic_net, and prox_nuclear, returns a numeric vector or matrix of class "numeric" or "matrix" (matching the shape and dimensions of input x) representing the evaluated proximal point operator. For grad_my_envelope, returns a numeric vector or matrix of class "numeric" or "matrix" containing the computed gradient of the Moreau-Yosida envelope at x.

References

Shukla A, Vats D, Chi EC (2025). "Proximal Hamiltonian Monte Carlo." *arXiv preprint*, doi:10.48550/arXiv.2510.22252.

---

specialized_models *Specialized High-Level Estimators Using Proximal HMC*

Description

Fits sparse logistic regression, nuclear-norm low-rank matrix recovery, and Bayesian Lasso linear regression models using the p-HMC algorithm, matching the case studies in Section VI of Shukla, Vats, and Chi (2025).

Usage

```
phmc_logistic(
  X,
  y,
  alpha = 1,
  lambda_g = 0.01,
  n_draws = 2000,
  epsilon = 0.002,
  L = 10,
  ...
)

phmc_matrix(
  X,
  alpha = 1,
  sigma_sq = 1,
  lambda_g = 1e-04,
  n_draws = 1000,
  epsilon = 0.001,
  L = 10,
  ...
)

phmc_lasso(
  X,
  y,
  alpha = 1,
  sigma_sq = 1,
  lambda_g = 0.01,
  n_draws = 2000,
  epsilon = 0.005,
  L = 10,
  ...
)
```

Arguments

X	Numeric design matrix (for regression models) or observed noisy matrix (for matrix recovery).
y	Numeric binary response vector for logistic regression or continuous response vector for Lasso.
alpha	Numeric scalar > 0. Regularization penalty parameter (default 1.0).
lambda_g	Numeric scalar > 0. Moreau-Yosida envelope regularization parameter.
n_draws	Integer > 0. Total number of MCMC iterations.
epsilon	Numeric scalar > 0. Step size parameter.
L	Integer >= 1. Number of leapfrog steps.

...	Additional arguments passed to phmc.
sigma_sq	Numeric scalar > 0. Error variance parameter (default 1.0).

Value

An object of class "phmc", which is a list containing posterior MCMC draws, parameter summary estimates matrix, log-likelihood, information criteria (AIC, BIC, DIC), acceptance rate, and execution metadata. See phmc for detailed descriptions of the list elements and output meaning.

References

Shukla A, Vats D, Chi EC (2025). "Proximal Hamiltonian Monte Carlo." *arXiv preprint*, doi:10.48550/arXiv.2510.22252.

Index