

Package: rvmprior (via r-universe)

July 22, 2026

Type Package

Title Relevance Vector Machine with Flexible Spike-and-Slab Priors

Version 0.1.0

Description Fits Relevance Vector Machine (RVM) models for Normal, Epanechnikov, and Bickel prior distributions, and Gaussian or Mexican Hat kernels.

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.0.0

Imports ggplot2, kernlab, MASS

NeedsCompilation no

Author Alex Sousa [aut, cre], Rafael Oliveira [aut]

Maintainer Alex Sousa <asousa@unicamp.br>

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

Date/Publication 2026-07-22 07:30:02 UTC

RemoteUrl <https://github.com/cran/rvmprior>

RemoteRef HEAD

RemoteSha debf01dc77d4d5a646dbbbb362af211c5517122

Contents

plot.rvm_prior	2
rvm_prior	2
sinc_data	4

Index	5
--------------	----------

plot.rvm_prior	<i>Fit plot of an rvm_prior object</i>
----------------	--

Description

S3 visualization method for objects of class "rvm_prior". Displays the training data, the predicted curve, the true function (if available), and the selected relevance vectors.

Usage

```
## S3 method for class 'rvm_prior'
plot(x, f_true = NULL, ...)
```

Arguments

x	Object of class "rvm_prior".
f_true	Numeric vector or NULL. True function to overlay on the plot.
...	Additional arguments (ignored).

Value

Invisible ggplot object. The plot is printed as a side effect.

Examples

```
dados <- sinc_data(n = 100, SNR = 7)
fit <- rvm_prior(dados$x, dados$y, prior = "Epanechnikov",
                 kernel = "gaussian", f_true = dados$f_true)
plot(fit)
```

rvm_prior	<i>Fits an RVM model with flexible prior</i>
-----------	--

Description

Main function of the rvmprior package. Fits a Relevance Vector Machine (RVM) model for regression with support for three types of priors (Normal, Epanechnikov, Bickel) and two kernel types (gaussian, mexican_hat).

Usage

```
rvm_prior(  
  x,  
  y,  
  x_test = NULL,  
  prior = c("Normal", "Epanechnikov", "Bickel"),  
  kernel = c("gaussian", "mexican_hat"),  
  kernel_param = NULL,  
  prior_param = NULL,  
  alpha = 0.95,  
  show_plot = FALSE,  
  f_true = NULL,  
  ...  
)
```

Arguments

x	Numeric vector. Predictor variable (unidimensional).
y	Numeric vector. Response variable. Must have the same length as x.
x_test	Numeric vector or NULL. Prediction points. If NULL, uses x.
prior	Character. Prior for the weights: "Normal", "Epanechnikov", or "Bickel".
kernel	Character. Kernel function: "gaussian" or "mexican_hat".
kernel_param	Numeric or NULL. Kernel parameter. If NULL, uses an automatic value.
prior_param	Numeric or NULL. Prior parameter. If NULL, uses an automatic value.
alpha	Numeric between 0 and 1. Spike probability. Default: 0.95.
show_plot	Logical. If TRUE, displays the fit plot. Default: FALSE.
f_true	Numeric vector or NULL. True function for comparison in the plot.
...	Additional arguments passed to rvm_spike_slab (iterations, tol).

Value

Invisible object of class "rvm_prior" with the fit results.

Examples

```
dados <- sinc_data(n = 100, SNR = 7)  
fit <- rvm_prior(dados$x, dados$y, prior = "Epanechnikov", kernel = "gaussian")  
print(fit)
```

sinc_data	<i>Generate simulated data from the sinc function with Gaussian noise</i>
-----------	---

Description

The sinc function is defined as $\text{sinc}(x) = \sin(x)/x$, with limit $\text{sinc}(0) = 1$. Used in nonparametric regression simulations.

Usage

```
sinc_data(n = 100, from = -10, to = 10, SNR = NULL, sd_noise = 0.2, seed = 123)
```

Arguments

n	Integer. Number of points to generate. Default: 100.
from	Numeric. Lower bound of the x domain. Default: -10.
to	Numeric. Upper bound of the x domain. Default: 10.
SNR	Numeric or NULL. Desired signal-to-noise ratio. If NULL, uses sd_noise.
sd_noise	Numeric. Standard deviation of the noise. Used only when SNR is NULL. Default: 0.2.
seed	Integer. Seed for reproducibility. Default: 123.

Value

An object of class "sinc_data" with elements x, y, f_true, sd_noise and SNR.

Examples

```
dados <- sinc_data(n = 100, SNR = 7)
print(dados)
```

Index

`plot.rvm_prior`, [2](#)

`rvm_prior`, [2](#)

`sinc_data`, [4](#)