Package 'AddiVortes'

August 4, 2026

Title (Bayesian) Additive Voronoi Tessellations

Version 0.6.9

Description Implements the Bayesian Additive Voronoi Tessellation model for non-parametric regression and machine learning as introduced in Stone and Gosling (2025) <doi:10.1080/10618600.2024.2414104>. This package provides a flexible alternative to BART (Bayesian Additive Regression Trees) using Voronoi tessellations instead of trees. Users can fit Bayesian regression models (estimating the associated posterior distributions and make predictions. It is particularly useful for spatial data analysis, machine learning regression, complex function approximation and Bayesian modeling where the underlying structure is unknown. The method is well-suited to capturing spatial patterns and non-linear relationships.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.0.0)

Imports parallel, pbapply (>= 1.6)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr, xml2

SystemRequirements C++20

Config/testthat/edition 3

URL https://johnpaulgosling.github.io/AddiVortes/

VignetteBuilder knitr

BugReports https://github.com/johnpaulgosling/AddiVortes/issues

LazyData true

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation yes

Author Adam Stone [aut] (ORCID: <https://orcid.org/0009-0004-0058-6117>), John Paul Gosling [aut, cre] (ORCID: <https://orcid.org/0000-0002-4072-3022>), Andrew Iskauskas [aut] (ORCID: <https://orcid.org/0000-0003-2825-3651>)

Maintainer John Paul Gosling <john-paul.gosling@durham.ac.uk>

Repository CRAN

Date/Publication 2026-08-04 00:20:02 UTC

Contents

AddiVortes-package 2
AddiVortes 3
Boston 5
cellIndices 6
Cylinder 7
new_AddiVortes 7
plot.AddiVortes 8
predict.AddiVortes 10
print.AddiVortes 12
summary.AddiVortes 12
traceplots 13
traceplots.AddiVortes 13
Weather 14

Index **16**

`AddiVortes-package` *AddiVortes: Bayesian Additive Voronoi Tessellations for Machine Learning*

Description

AddiVortes implements Bayesian Additive Voronoi Tessellation models for machine learning regression and non-parametric statistical modeling. This package provides a flexible alternative to BART (Bayesian Additive Regression Trees), using Voronoi tessellations instead of trees for spatial partitioning. The method is particularly effective for spatial data analysis, complex function approximation, and Bayesian regression modeling.

Details

Key features include:

- Machine learning regression with Bayesian inference
- Alternative to BART using Voronoi tessellations
- Spatial data analysis and modeling
- Non-parametric regression capabilities
- Complex function approximation
- Uncertainty quantification through posterior inference

Author(s)

Maintainer: John Paul Gosling <john-paul.gosling@durham.ac.uk> (ORCID)

Authors:

- Adam Stone <adam.stone2@durham.ac.uk> (ORCID)
- Andrew Iskauskas <andrew.iskauskas@durham.ac.uk> (ORCID)

References

Stone, A. and Gosling, J.P. (2025). AddiVortes: (Bayesian) additive Voronoi tessellations. Journal of Computational and Graphical Statistics.

See Also

https://johnpaulgosling.github.io/AddiVortes/

`AddiVortes` *Fit an AddiVortes regression model*

Description

The AddiVortes model is a Bayesian nonparametric regression model that uses a tessellation to model the relationship between the covariates and the output values. The model uses a backfitting algorithm to sample from the posterior distribution of the output values for each tessellation. Alongside fitting details and the posterior sample, the function returns the RMSE value for the test samples.

The function can handle multiple types of covariates, including continuous, spherical and categorical. Categorical covariates are automatically detected and one-hot encoded, with the first level of each categorical variable used as the reference category. The `catScaling` parameter allows control over the weight of categorical differences in distance calculations. For spherical covariates, the function assumes that the final spherical dimension corresponds to the polar angle, which has a range of 0 to 2*pi. The `metric` parameter can be used to specify the type of each covariate (Euclidean, Spherical, or Categorical), and the `members` parameter can indicate membership of covariates into different subspaces when using multiple spheres in covariate space.

Usage

```
AddiVortes(
  y,
  x,
  m = 200,
  totalMCMCIter = 1200,
  mcmcBurnIn = 200,
  nu = 6,
  q = 0.85,
  k = 3,
```

```
  sd = 0.8,
  Omega = min(3, ncol(x)),
  LambdaRate = 5,
  InitialSigma = "Linear",
  thinning = 1,
  metric = "E",
  members = NULL,
  catScaling = 1,
  cat.onehot = TRUE,
  showProgress = interactive()
)
```

Arguments

y	A vector of the output values.
x	A matrix or data frame of the covariates. Character and factor columns are treated as categorical variables and automatically converted to d-1 binary indicator variables via one-hot encoding (with the first level as reference).
m	The number of tessellations.
totalMCMCIter	The number of iterations.
mcmcBurnIn	The number of burn in iterations.
nu	The degrees of freedom.
q	The quantile.
k	The number of centres.
sd	The standard deviation used in centre proposals.
Omega	Omega/(number of covariates) is the prior probability of adding a dimension.
LambdaRate	The rate of the Poisson distribution for the number of centres.
InitialSigma	The method used to calculate the initial variance.
thinning	The thinning rate.
metric	Either "E" (Euclidean, default), "S" (Spherical), or "C" (Categorical).
members	If needed, indicates membership of covariates into different subspaces (needed if using multiple spheres in covariate space). Default NULL.
catScaling	Numeric scalar controlling the scale of binary indicator variables created from categorical covariates. Each binary indicator takes values 0 (reference level) or catScaling (non-reference level). The default value of 1 matches the range of continuous covariates, which are normalised to [-0.5, 0.5] (range = 1) during fitting, so categorical differences receive comparable weight to continuous differences in the distance calculations. Increase above 1 to give categorical differences more weight; decrease below 1 to give them less weight. Binary indicator columns are named <colname>_<level> (e.g. a column grp with levels "A", "B", "C" produces columns grp_B and grp_C, with "A" as the reference level).
cat.onehot	Should categorical covariates be one-hot encoded? Default TRUE.
showProgress	Logical; if TRUE, progress bars and messages are shown during fitting.

Value

An AddiVortes object containing the posterior samples of the tessellations, dimensions and predictions, plus per-iteration trace statistics used by `traceplots()`.

Examples

```
# Simple example with simulated data
set.seed(123)
x <- matrix(rnorm(50), 10, 5)
y <- rnorm(10)
# Fit model with reduced iterations for quick example
fit <- AddiVortes(y, x, m = 5, totalMCMCIter = 50, mcmcBurnIn = 10)

# Larger example with categorical covariates (d=2 and d=3) and a test set
set.seed(456)
n_train <- 200
n_test <- 50
x_train <- data.frame(
  x1   = rnorm(n_train),
  x2   = runif(n_train),
  grp2 = sample(c("A", "B"), n_train, replace = TRUE),
  grp3 = sample(c("low", "mid", "high"), n_train, replace = TRUE)
)
y_train <- x_train$x1 + ifelse(x_train$grp2 == "B", 1, 0) + rnorm(n_train, sd = 0.5)

fit2 <- AddiVortes(y_train, x_train,
  m = 10, totalMCMCIter = 200, mcmcBurnIn = 50,
  catScaling = 1, showProgress = FALSE
)

x_test <- data.frame(
  x1   = rnorm(n_test),
  x2   = runif(n_test),
  grp2 = sample(c("A", "B"), n_test, replace = TRUE),
  grp3 = sample(c("low", "mid", "high"), n_test, replace = TRUE)
)
y_test <- x_test$x1 + ifelse(x_test$grp2 == "B", 1, 0) + rnorm(n_test, sd = 0.5)

preds <- predict(fit2, x_test, showProgress = FALSE)
test_rmse <- sqrt(mean((y_test - preds)^2))
```

Boston *Boston Dataset*

Description

The Boston Housing Dataset, derived from information from the U.S. Census Service.

Usage

```
Boston
```

Format

`Boston`**:**

A data.frame with 506 rows and 14 columns:

x1-x13 The covariates in the data

y The response variable

Source

Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol. 5, 81-102, 1978

`cellIndices` *Assign Observations to Tessellation Cells*

Description

For a given tessellation, this function identifies which cell (centre) each observation belongs to based on nearest neighbour classification.

Usage

```
cellIndices(x, tess, dim, metric = "E", members)
```

Arguments

`x`	A numeric matrix of covariates where each row is an observation.
`tess`	A numeric matrix representing the tessellation centres, where each row is a unique centre.
`dim`	An integer vector specifying the column indices of `x` to be used for calculating distance.
`metric`	Either "Euclidean" or "Spherical".
`members`	A vector indicating covariate membership.

Details

It finds the closest tessellation centre for each observation (row) in the covariate matrix, considering only the specified dimensions. This is achieved using the k-nearest neighbour algorithm where k=1.

Value

A numeric vector of integers where each element corresponds to a row in `x` and its value is the row index of the nearest centre in `tess`.

Cylinder *Cylindrical Dataset*

Description

Synthetic data generated for cylindrical parameters. The response is given by the function f(z, theta) = sin(z)cos(theta) + z*sin(theta)^2, with added noise.

Usage

```
Cylinder
```

Format

`Cylinder`**:**
A data.frame with 400 rows and 3 columns:

z Height

theta Polar angle

Y Response

new_AddiVortes *Create an AddiVortes Object*

Description

A constructor for the AddiVortes class.

Usage

```
new_AddiVortes(
  posteriorTess,
  posteriorDim,
  posteriorSigma,
  posteriorPred,
  xCentres,
  xRanges,
  yCentre,
  yRange,
  inSampleRmse,
  metric = "E",
  members = rep(1, length(xCentres)),
  metric_aug = "E",
  member_aug = rep(1, length(xCentres)),
  catEncoding = NULL,
  traceStats = NULL
)
```

Arguments

posteriorTess	A list of the posterior samples of the tessellations.
posteriorDim	A list of the posterior samples of the dimensions.
posteriorSigma	A list of the posterior samples of the error variance.
posteriorPred	A list of the posterior samples of the predictions.
xCentres	The centres of the covariates.
xRanges	The ranges of the covariates.
yCentre	The centre of the output values.
yRange	The range of the output values.
inSampleRmse	The in-sample RMSE.
metric	The metric used for scaling covariates (default "E" for Euclidean).
members	The membership vector for the covariates
metric_aug	The augmented metric after categorical variables are converted to one-hot
member_aug	The membership vector corresponding to metric_aug
catEncoding	Optional list of categorical encoding metadata returned by encodeCategories_internal, or NULL if no categorical covariates were present.
traceStats	Optional data frame of per-iteration MCMC trace statistics.

Value

An object of class AddiVortes.

---

plot.AddiVortes *Plot Method for AddiVortes*

Description

Generates comprehensive diagnostic plots for a fitted AddiVortes object. This function creates multiple diagnostic plots including residuals, MCMC traces for sigma, and tessellation complexity over iterations.

Usage

```
## S3 method for class 'AddiVortes'
plot(
  x,
  x_train,
  y_train,
  sigma_trace = NULL,
  which = c(1, 2, 3),
  ask = FALSE,
  ...
)
```

Arguments

x	An object of class AddiVortes, typically the result of a call to AddiVortes().
x_train	A matrix of the original training covariates.
y_train	A numeric vector of the original training true outcomes.
sigma_trace	An optional numeric vector of sigma values from MCMC samples. If not provided, the method will attempt to extract the posterior error standard deviation from the model object.
which	A numeric vector specifying which plots to generate: 1 = Residuals plot, 2 = Sigma trace, 3 = Tessellation complexity trace, 4 = Predicted vs Observed. Default is c(1, 2, 3).
ask	Logical; if TRUE, the user is asked to press Enter before each plot.
...	Additional arguments passed to plotting functions.

Details

The function generates up to four diagnostic plots:

1. **Residuals Plot**: Residuals vs fitted values with smoothed trend line
2. **Sigma Trace**: MCMC trace plot for the error standard deviation
3. **Tessellation Complexity**: Trace of average tessellation size over iterations
4. **Predicted vs Observed**: Scatter plot with credible intervals

Value

This function is called for its side effect of creating plots and returns NULL invisibly.

Examples

```
## Not run:
# Assuming 'fit' is a trained AddiVortes object
plot(fit, x_train = x_train_data, y_train = y_train_data)

# Show only specific plots
plot(fit, x_train = x_train_data, y_train = y_train_data, which = c(1, 3))

# With custom sigma trace
plot(fit,
  x_train = x_train_data, y_train = y_train_data,
  sigma_trace = my_sigma_samples
)

## End(Not run)
```

`predict.AddiVortes` *Predict Method for AddiVortes*

Description

Predicts outcomes for new data using a fitted `AddiVortes` model object. It can return mean predictions, quantiles and optionally calculate the Root Mean Squared Error (RMSE) if true outcomes are provided.

Usage

```
## S3 method for class 'AddiVortes'
predict(
  object,
  newdata,
  type = c("response", "quantile"),
  quantiles = c(0.025, 0.975),
  interval = c("credible", "prediction"),
  showProgress = interactive(),
  parallel = TRUE,
  cores = NULL,
  ...
)
```

Arguments

`object`	An object of class `AddiVortes`, typically the result of a call to `AddiVortes()`.
`newdata`	A matrix of covariates for the new test set. The number of columns must match the original training data.
`type`	The type of prediction required. The default `"response"` gives the mean prediction. The alternative `"quantile"` returns the quantiles specified by the `quantiles` argument.
`quantiles`	A numeric vector of probabilities to compute for the predictions when `type = "quantile"`.
`interval`	The type of interval calculation. The default `"credible"` accounts only for uncertainty in the mean (similar to `lm`'s confidence interval). The alternative `"prediction"` also includes the model's error variance, producing wider intervals (similar to `lm`'s prediction interval).
`showProgress`	Logical; if TRUE, a progress bar is shown during prediction.
`parallel`	Logical; if TRUE (default), predictions are computed in parallel.
`cores`	The number of CPU cores to use for parallel processing. If NULL (default), it defaults to one less than the total number of available cores.
`...`	Further arguments passed to or from other methods (currently unused).

Details

This function relies on the internal helper function `applyScaling_internal` being available in the environment, which is used by the main `AddiVortes` function.

When `interval = "prediction"` and `type = "quantile"`, the function samples additional Gaussian noise with variance equal to the sampled sigma squared from the posterior. This accounts for the inherent variability in individual predictions, not just uncertainty in the mean function. The noise is added in the scaled space before unscaling predictions.

Value

If `type = "response"`, a numeric vector of mean predictions. If `type = "quantile"`, a matrix where each row corresponds to an observation in `newdata` and each column to a quantile.

Examples

```
# Fit a model
set.seed(123)
X <- matrix(rnorm(100), 20, 5)
Y <- rnorm(20)
fit <- AddiVortes(Y, X, m = 5, totalMCMCIter = 50, mcmcBurnIn = 10)

# New data for prediction
X_new <- matrix(rnorm(25), 5, 5)

# Mean predictions
pred_mean <- predict(fit, X_new, type = "response")

# Credible intervals (uncertainty in mean only)
pred_conf <- predict(fit, X_new,
  type = "quantile",
  interval = "credible",
  quantiles = c(0.025, 0.975)
)

# Prediction intervals (includes error variance)
pred_pred <- predict(fit, X_new,
  type = "quantile",
  interval = "prediction",
  quantiles = c(0.025, 0.975)
)

# Prediction intervals are wider than credible intervals
mean(pred_pred[, 2] - pred_pred[, 1]) > mean(pred_conf[, 2] - pred_conf[, 1])
```

print.AddiVortes *Print Method for AddiVortes*

Description

Prints a summary of a fitted `AddiVortes` object, providing information about the model structure, dimensions, and fit quality similar to the output of a linear model summary.

Usage

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

Arguments

`x`	An object of class `AddiVortes`, typically the result of a call to `AddiVortes()`.
`...`	Further arguments passed to or from other methods (currently unused).

Details

The print method displays:

- The model formula representation
- Number of covariates and posterior samples
- Number of tessellations used
- In-sample RMSE
- Covariate scaling information

Value

The function is called for its side effect of printing model information and returns the input object `x` invisibly.

summary.AddiVortes *Summary Method for AddiVortes*

Description

Provides a detailed summary of a fitted `AddiVortes` object, including more comprehensive information than the print method.

Usage

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

Arguments

`object`	An object of class `AddiVortes`, typically the result of a call to `AddiVortes()`.
`...`	Further arguments passed to or from other methods (currently unused).

Value

The function is called for its side effect of printing detailed model information and returns the input object `object` invisibly.

`traceplots` *Trace Plot Diagnostics*

Description

Creates trace plots for MCMC diagnostics of a fitted model.

Usage

```
traceplots(x, ...)
```

Arguments

`x`	A fitted model object.
`...`	Further arguments passed to methods.

Value

This function is called for its side effect of creating plots and returns `NULL` invisibly.

`traceplots.AddiVortes` *Trace Plot Diagnostics for AddiVortes*

Description

Displays four MCMC trace plots for a fitted `AddiVortes` object: the average number of centres per tessellation, the standard deviation of the number of centres per tessellation, the average number of dimensions used per tessellation, and the retained-state log-likelihood component whose differences form the likelihood part of the acceptance ratio.

Usage

```
## S3 method for class 'AddiVortes'
traceplots(x, ask = FALSE, ...)
```

Arguments

x	An object of class AddiVortes, typically the result of a call to AddiVortes().
ask	Logical; if TRUE, the user is asked to press Enter before each plot.
...	Additional arguments passed to plotting functions.

Details

The four trace plots are:

1. **Average Centres**: Average number of centres per tessellation.
2. **Centre Count Standard Deviation**: Standard deviation of the number of centres per tessellation.
3. **Average Dimensions**: Average number of active dimensions used per tessellation.
4. **Log Likelihood**: Average retained-state log-likelihood component at the end of each MCMC iteration.

Value

This function is called for its side effect of creating plots and returns NULL invisibly.

Examples

```
## Not run:
# Assuming 'fit' is a trained AddiVortes object
traceplots(fit)

## End(Not run)
```

Weather *Weather Dataset*

Description

A subset of data collected as part of the GES DISC datasets for calibrated brightness temperatures.

Usage

```
Weather
```

Format

Weather**:**

A data.frame with 2000 rows and 3 columns:

theta Polar angle

phi Azimuthal angle

Y Response variable

Source

https://disc.gsfc.nasa.gov/datasets/GPM_1CGPMGMI_07/summary?keywords=10.5067%2FGPM%2FGMI%2FGPM%2F1

Index

∗ **BART**
    AddiVortes-package, 2
∗ **bayesian**
    AddiVortes-package, 2
∗ **datasets**
    Boston, 5
    Cylinder, 7
    Weather, 14
∗ **machine-learning**
    AddiVortes-package, 2
∗ **package**
    AddiVortes-package, 2
∗ **regression**
    AddiVortes-package, 2
∗ **spatial**
    AddiVortes-package, 2
∗ **tessellation**
    AddiVortes-package, 2

AddiVortes, 3
AddiVortes-package, 2

Boston, 5

cellIndices, 6
Cylinder, 7

new_AddiVortes, 7

plot.AddiVortes, 8
predict.AddiVortes, 10
print.AddiVortes, 12

summary.AddiVortes, 12

traceplots, 13
traceplots.AddiVortes, 13

Weather, 14