

Package ‘gdam’

September 24, 2026

Title Fast Robust Additive Models using Gamma Divergence

Version 0.0.1

Date 2026-09-10

Description Provides a computationally efficient method (at least at the time of development!) for robust additive modeling based on the gamma divergence, implementing the methodology of Hui, Ding, and Sugasawa (2026) <[doi:10.1007/s11222-026-10960-3](https://doi.org/10.1007/s11222-026-10960-3)>. The method assumes an identity link and normally distributed errors, and applies the gamma divergence to the resulting restricted maximum likelihood function so as to obtain a loss function that is less sensitive to outlying responses. Because 'gdam' uses a model fitted via the 'mgcv' package as the starting point for optimization, and leverages many of its existing techniques for estimation and inference, users can take advantage of many of the smoothing options available in the 'mgcv' package for constructing additive models.

Maintainer Francis KC Hui <fhui28@gmail.com>

License GPL (>= 3)

Encoding UTF-8

Imports colorspace, ggplot2, gratia, Matrix, mgcv, stats

Suggests spelling, tidyverse

Language en-US

URL <https://github.com/fhui28/gdam>

BugReports <https://github.com/fhui28/gdam/issues>

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation no

Author Francis KC Hui [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0765-3533>>),
Shonosuke Sugasawa [ctb] (ORCID:
<<https://orcid.org/0000-0002-9495-4280>>)

Repository CRAN

Date/Publication 2026-09-24 13:30:20 UTC

Contents

add_partial_residuals.gdam	2
appraise.gdam	3
gdam	5
predict.gdam	9
print.gdam	10
residuals.gdam	11
smooth_estimates	12
summary.gdam	13
Index	15

add_partial_residuals.gdam

Add partial residuals from a fitted gdam model

Description

[Experimental]

This function is basically a wrapper around the [gratia::add_partial_residuals\(\)](#) function for adding partial residuals, but designed to work with gdam objects. Please see the associated help file in the gratia package for more details.

Usage

```
## S3 method for class 'gdam'
add_partial_residuals(data, model, select = NULL, partial_match = FALSE, ...)
```

Arguments

data	A data frame containing values for the variables used to fit the gdam model.
model	A fitted gdam object.
select	Character, logical, or numeric; which smooths to plot; see gratia::add_partial_residuals() for details.
partial_match	Should smooths be selected by partial matches with select? If TRUE, select can only be a single string to match against.
...	Arguments passed to other methods.

Value

A data frame of the same dimension as data with partial residuals added on, while the final column also contains the final weights from the MM algorithm used to fit the gdam model.

Note

Acknowledgments to Gavin Simpson and the gratia package for the original [gratia::add_partial_residuals\(\)](#) function.

Author(s)Francis K.C. Hui fhui28@gmail.com

appraise.gdam

*Model diagnostics plots for gdam objects***Description****[Experimental]**

This function is basically a wrapper around the `gratia::appraise()` function, but designed to work with gdam objects. The main difference is that observations (residuals, quantiles etc...) in the diagnostic plots are colored by the final weights from the gdam object.

Usage

```
## S3 method for class 'gdam'
appraise(
  model,
  use_worm = FALSE,
  n_uniform = 10,
  n_simulate = 50,
  method = "uniform",
  type = "deviance",
  n_bins = "sturges",
  ncol = NULL,
  nrow = NULL,
  guides = "keep",
  level = 0.9,
  ci_col = "black",
  ci_alpha = 0.2,
  line_col = "red",
  ...
)
```

Arguments

<code>model</code>	A fitted gdam object.
<code>use_worm</code>	Logical. Should the worm plot be drawn in place of the QQ plot? Default is FALSE.
<code>n_uniform</code>	Integer. The number of times to randomize uniform quantiles in the direct computation method or QQ plots.
<code>n_simulate</code>	Integer. The number of times to simulate from the estimated model in the simulation-based method for QQ plots.
<code>method</code>	Character. The method to use for to generate theoretical quantiles. Options are "uniform" or "simulate".

type	Character. The type of residuals to use. Options are "deviance" or "pearson" residuals.
n_bins	Character. The number of bins to use in the histogram of residuals. See gratia::appraise() for more details.
ncol	Integer. The number of columns in the grid of plots. If NULL, the number of columns is determined automatically.
nrow	Integer. The number of rows in the grid of plots. If NULL, the number of rows is determined automatically.
guides	Character. The guide to use for the color scale. See gratia::appraise() for more details
level	Numeric. The confidence level for the reference intervals in the QQ plot. Only used if method = "simulate".
ci_col	Character. The color of the reference interval lines in the QQ plot. Only used if method = "simulate".
ci_alpha	Numeric. The alpha transparency level of the reference interval lines in the QQ plot. Only used if method = "simulate".
line_col	Character. The color of the reference 1:1 line in the QQ plot, and the reference line in the residuals vs. linear predictor plot.
...	Currently not used.

Value

A set of model diagnostic plots as per the original [gratia::appraise\(\)](#) function.

Note

Acknowledgments to Gavin Simpson and the [gratia](#) package for the original [gratia::appraise\(\)](#) function.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples.
```

Description

[Experimental]

This function fits a robust additive model, assuming an identity link function and a normal distribution for the errors, using a gamma divergence adaption of the restricted maximum likelihood function (REML) as the loss function (hence the name gdam, which stands for gamma divergence additive models).

The functions works by taking an object of class `mgcv::gam()` and uses it as a starting point for optimization of the gamma divergence, which itself is performed via a minorization-maximization (MM) algorithm.

Usage

```
gdam(
  gamObject,
  gamma_tuning = 0.3,
  control = list(tol = 1e-06, max_iteration = 1000, rinse_and_repeat = 0)
)
```

Arguments

- | | |
|--------------|--|
| gamObject | An object of class <code>mgcv::gam()</code> fitted using the mgcv package. It is strongly recommended that the additive model is fitted using <code>method = REML</code> . |
| gamma_tuning | A positive tuning parameter for the gamma divergence, and typically less than 0.5 for almost most additive models. The closer this value is to zero, the more the gamma divergence behaves like standard restricted maximum likelihood estimation i.e., the same as what is done in mgcv. |
| control | A list of control parameters for the MM algorithm. <ul style="list-style-type: none"> • <code>tol</code> The tolerance for convergence of the MM algorithm. Default is 1e-6. • <code>max_iteration</code> The maximum number of iterations for the MM algorithm. Default is 1000. • <code>rinse_and_repeat</code> An integer value for the number of times to perform the entire MM algorithm, but with a refitted GAM using the final weights from the previous application of the MM algorithm. This tends to be useful for more complex smooths like tensor product smooths, where the MM algorithm can sometimes get stuck in local maxima. Default is 0. |

Details

See manuscript for more details (currently in preparation).

Value

An object of class `gdam` with the following components:

- `call` The matched call.
- `coefficients` The estimated (smoothing) coefficients of the additive model.
- `sp` The estimated smoothing parameters of the additive model.
- `sigma2` The estimated residual variance of the additive model.
- `linear_predictors` The linear predictors of the additive model.
- `fitted_values` The fitted values of the additive model.
- `residuals` The raw residuals of the additive model.
- `final_weights` The final weights from the MM algorithm. The weights sum to the total number of observations in the data. Weights comparably less than 1 indicate that the observation is down-weighted in the estimation process i.e., is considered more likely to be an outlying observation
- `gamma_tuning` The gamma tuning parameter used in the estimation process.
- `gamma_divergence` The value of the gamma divergence at convergence.
- `gamObject` The original `gamObject` used to fit the additive model.
- `covariance_matrix` The estimated covariance matrix of the parameters, including of the residual variance (but not the smoothing parameters).
- `Hscore` The conditional Hyvarinen score (H-score) of the additive model. This is a measure that can be used to select the tuning parameter `gamma_tuning`, when comparing across difference `gdam` fits. The H-score calculation is based on the proposal of Sugusawa and Yonekura (2021) [doi:10.3390/e23091147](https://doi.org/10.3390/e23091147).
- `aic` The conditional Akaike Information Criterion (AIC) of the additive model. This is calculated as $-2 \times \text{gamma_divergence} + 2 \times \text{edf}$, where `edf` is the effective degrees of freedom of the additive model. Effectively adapts the idea of Kurara (2024) [doi:10.1080/03610926.2022.2155788](https://doi.org/10.1080/03610926.2022.2155788); see also `mgcv::logLik.gam()`.

Note**Warning**

Not all smooth specifications available in `mgcv` are supported let alone being properly tested e.g., we don't think smooths linked using the "id" argument, linear functional terms, or varying coefficient models as available using the "by" argument are supported; see `mgcv::gam.models()` for specification of GAMs. Apologies in advance if you encounter any issues with these, but do email the author if encounter any issues or have a feature request.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
library(mgcv)
library(tidyverse)
library(gratia)

# Simulate a additive model and add some outliers.
set.seed(062025)
simdat <- gamSim(eg = 7,
  n = 200,
  dist = "normal",
  scale = 0.5,
  verbose = TRUE)

create_outliers <- rbinom(nrow(simdat), size = 1, prob = 0.05)
# Could also the probability homogeneous e.g., vary with one or more covariate values.
table(create_outliers)
simdat$contaminated_y <- simdat$y
simdat$contaminated_y[create_outliers == 1] <- rnorm(sum(create_outliers == 1),
  mean = simdat$f[create_outliers==1],
  sd = 9)

# Fit the model using mgcv and then fit the robust additive model using gdam
fit_mgcv <- gam(contaminated_y ~ s(x0) + s(x1) + s(x2) + s(x3),
  data = simdat,
  family = gaussian(),
  method = "REML")

fit_gdam <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.1)
fit_gdam

summary(fit_gdam)

# Basic illustration of functions
residuals(object = fit_gdam)

appraise(fit_gdam)

## Automated partial smooth plots
gdam::smooth_estimates(fit_gdam) %>%
  gratia::draw() & theme_bw()

## A bit more manual but customized partial smooth plots -- all covariates
gdam::smooth_estimates(fit_gdam) %>%
  gratia::add_confint() %>%
  pivot_longer(x0:x3, values_to = "x", names_to = "covariate") %>%
  filter(!is.na(x)) %>%
  ggplot() +
  geom_line(aes(x = x, y = .estimate), lwd = 1.2) +
  geom_ribbon(aes(ymin = .lower_ci, ymax = .upper_ci, x = x), alpha = 0.2) +
  geom_rug(aes(x = x),
  data = simdat %>% pivot_longer(x0:x3, values_to = "x", names_to = "covariate"),
```

```

sides = "b") +
facet_wrap(. ~ covariate, scale = "free", nrow = 2) +
theme_bw()

## A bit more manual but customized partial smooth plots -- one covariate
## but adding partial residuals
simdat_expanded <- add_partial_residuals(data = simdat, model = fit_gdam, select = NULL)
gdam::smooth_estimates(fit_gdam) %>%
gratia::add_confint() %>%
filter(.smooth == s(x0)) %>%
ggplot() +
geom_point(aes(x = x0, y = `s(x0)`, color = weights), data = simdat_expanded, cex = 1.5) +
geom_line(aes(x = x0, y = .estimate), lwd = 1.2) +
geom_ribbon(aes(ymin = .lower_ci, ymax = .upper_ci, x = x0), alpha = 0.2) +
scale_color_viridis_c() +
theme_bw()

predict(fit_gdam, se.fit = TRUE)

# Simple check of the in-sample performance.
# Please see the associated manuscript for a more systematic evaluation of
# gdam including against other robust methods for additive models
cbind(simdat$f, fit_mgcv$fitted.values, fit_gdam$fitted.values) %>%
t %>%
dist()

# Checking various selection criteria -- mainly for choosing the gamma tuning parameter
# Personally we prefer using the H-score for this as it is slightly more
# principled in its construction for gamma divergence
fit_gdam00 <- gdam(gamObject = fit_mgcv, gamma_tuning = 1e-6)
fit_gdam001 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.01)
fit_gdam01 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.1)
fit_gdam02 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.2)
fit_gdam03 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.3)
fit_gdam04 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.4)
fit_gdam05 <- gdam(gamObject = fit_mgcv, gamma_tuning = 0.5)

fit_gdam00$Hscore
fit_gdam001$Hscore
fit_gdam01$Hscore
fit_gdam02$Hscore
fit_gdam03$Hscore
fit_gdam04$Hscore
fit_gdam05$Hscore

```

predict.gdam

*Predictions from a fitted gdam model.***Description****[Experimental]**

Takes a fitted gam object produced by `gdam()` and produces predictions given a new set of values for the model covariates or the original values used for the model fit. Predictions can be accompanied by standard errors and uncertainty intervals based on a large sample normality assumption for the distribution of the model coefficients. The routine can optionally return the matrix by which the model coefficients must be premultiplied in order to yield the values of the linear predictor.

Usage

```
## S3 method for class 'gdam'
predict(
  object,
  newdata,
  type = "response",
  se.fit = FALSE,
  coverage = 0.95,
  na.action = na.pass,
  ...
)
```

Arguments

<code>object</code>	A fitted gdam object.
<code>newdata</code>	A data frame containing the values of the covariates at which predictions are required. If <code>newdata</code> is omitted, predictions are made at the original data used to fit the model.
<code>type</code>	The type of prediction required. The default is on the scale of the linear predictor; the alternative response is on the scale of the response variable...which in the case of the additive models i.e., identity link function, yields the same predictions; the alternative <code>lpmatrix</code> returns the matrix by which the model coefficients must be pre-multiplied in order to yield the values of the linear predictor (in this case <code>se.fit</code> is ignored).
<code>se.fit</code>	Logical. Should standard errors be returned? Default is FALSE.
<code>coverage</code>	Numeric. The coverage of the uncertainty intervals. Default is 0.95.
<code>na.action</code>	A function which indicates what should happen when <code>newdata</code> contain NA values. The default is <code>na.pass</code> ; see <code>mgcv::predict.gam()</code> for more details.].
<code>...</code>	Currently not used.

Value

A vector of predictions, or a data frame containing the predictions, standard errors, and uncertainty intervals.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples.
```

print.gdam	<i>Print a fitted gdam model.</i>
------------	-----------------------------------

Description**[Experimental]**

This function prints out some details of the fitted gdam object in a human-readable format, similar to that of `mgcv::print.gam()`. This includes the tuning parameter used, the formula, the estimated degrees of freedom, and the gamma divergence. Note the **estimated degrees of freedom should be taken with a grain of salt**, as they are estimated from a post-hoc GAM fit (we conjecture that they are likely to be marginally underestimated).

Usage

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

Arguments

x	A fitted gdam object.
...	Currently not used.

Value

Invisible output object.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples.
```

residuals.gdam*Residuals from a fitted gdam model.*

Description

[Experimental]

Computes residuals from a fitted gdam object. The residuals can be of different types: "response" corresponds to the raw residuals, "deviance" corresponds to the deviance residuals (note these are slightly ad-hoc as the concept of deviance itself is not clearly defined in the case of gamma divergence; here the deviance residual is computed assuming normally distributed errors), "pearson" corresponds to the Pearson residuals, "scaled_pearson" corresponds to the Pearson residuals scaled by the square root of the estimated dispersion parameter, and "working" corresponds to the working residuals (i.e., the residuals from a post-hoc GAM fit, though in this case they should correspond to the raw response residuals).

Usage

```
## S3 method for class 'gdam'
residuals(object, type = "deviance", ...)
```

Arguments

object	A fitted gdam object.
type	The type of residuals to compute. Options are "response", "deviance", "pearson", "scaled_pearson", or "working".
...	Currently not used.

Value

A vector of residuals.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples.
```

smooth_estimates

*Evaluate smooths from a fitted gdam model at covariate values***Description****[Experimental]**

This function is basically a wrapper around the `gratia::smooth_estimates()` function for evaluating a smooth at a grid of evenly spaced value over the range of the covariate, but designed to work with gdam objects. Please see the associated help file in the `gratia` package for more details.

Usage

```
smooth_estimates(object, ...)
```

```
## S3 method for class 'gdam'
smooth_estimates(
  object,
  select = NULL,
  n = 100,
  n_3d = 16,
  n_4d = 4,
  data = NULL,
  dist = NULL,
  partial_match = FALSE,
  ...
)
```

Arguments

<code>object</code>	A fitted gdam object.
<code>...</code>	Currently not used.
<code>select</code>	A character vector of smooths to evaluate. If <code>NULL</code> , all smooths are evaluated.
<code>n</code>	The number of points at which to evaluate the smooths.
<code>n_3d</code>	The number of points over the range of the last covariate must in a 3D plot; see gratia::smooth_estimates() for details.
<code>n_4d</code>	The number of points over the range of the last covariate in a 4D plot; see gratia::smooth_estimates() for details.
<code>data</code>	A data frame containing the values of the covariates at which to evaluate the smooths. If <code>NULL</code> , the original data is used.
<code>dist</code>	Numeric; if greater than zero, this is used to determine when a location is too far from data to be plotted when plotting 2D smooths; see gratia::smooth_estimates() for details.
<code>partial_match</code>	Logical. In the case of character <code>select</code> , should <code>select</code> match partially against smooths? If <code>TRUE</code> then <code>select</code> only be a single string.

Value

A data frame which is of class `smooth_estimates`.

Note

Acknowledgments to Gavin Simpson and the `gratia` package for the original `gratia::smooth_estimates()` function.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples, particularly for constructing partial smooth plots.
```

summary.gdam

Summary for a fitted gdam model.

Description**[Experimental]**

Takes a fitted `gdam` object and produces some useful *post-hoc* summaries of the fitted model. Note all results **should be taken with a grain of salt**, as they are estimated from a post-hoc GAM fit using `mgcv::gam()` (we conjecture quantities like degrees of freedom are likely to be marginally underestimated, and hence P-values are likely to be marginally smaller than they should be). Please see `mgcv::summary.gam()` for details on the output.

Usage

```
## S3 method for class 'gdam'
summary(
  object,
  digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

<code>object</code>	A fitted <code>gdam</code> object.
<code>digits</code>	The number of significant digits to print.
<code>signif.stars</code>	If TRUE, p-values are printed with stars indicating significance.
<code>...</code>	Currently not used.

Value

A summary of the fitted gdam object, similar to that of [mgcv::summary.gam\(\)](#). Please see the help file for that function for more details.

Note**Warning**

The p-values are approximate and neglect smoothing parameter uncertainty. They are likely to be somewhat too low when smoothing parameter estimates are highly uncertain; please see the [mgcv::summary.gam\(\)](#) help file for more details.

Author(s)

Francis K.C. Hui fhui28@gmail.com

Examples

```
# See the main `gdam` help file for examples.
```

Index

`add_partial_residuals.gdam`, [2](#)
`appraise.gdam`, [3](#)

`gdam`, [5](#)
`gdam()`, [9](#)
`gratia::add_partial_residuals()`, [2](#)
`gratia::appraise()`, [3](#), [4](#)
`gratia::smooth_estimates()`, [12](#), [13](#)

`mgcv::gam()`, [5](#), [13](#)
`mgcv::gam.models()`, [6](#)
`mgcv::logLik.gam()`, [6](#)
`mgcv::predict.gam()`, [9](#)
`mgcv::print.gam()`, [10](#)
`mgcv::summary.gam()`, [13](#), [14](#)

`predict.gdam`, [9](#)
`print.gdam`, [10](#)

`residuals.gdam`, [11](#)

`smooth_estimates`, [12](#)
`summary.gdam`, [13](#)