Package: greta.gam (via r-universe)

July 8, 2024

Type Package

Title Generalised Additive Models in greta using mgcv

Version 0.1.0

Author c(person(``Nick'', ``Golding'', role = c(``aut'', ``cre''), email = ``nick.golding.research@gmail.com''), person(``David'', ``L'', ``Miller'', role = c(``aut'', ``cre''), email = ``dave@ninepointeightone.net''),

Maintainer Nick Golding <nick.golding.research@gmail.com>

Description A module for greta that lets you use mgcv's smoother functions and formula syntax to define smooth terms for use in a greta model. You can then define your own likelihood to complete the model, and fit it by MCMC.

License GPL(>=2)

Imports mgcv

Depends greta

Suggests testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/testthat/edition 3

Repository https://greta-dev.r-universe.dev

RemoteUrl https://github.com/greta-dev/greta.gam

RemoteRef HEAD

RemoteSha 402bbc9659fb5ac121c50a886746c36383082008

Contents

`evaluate_smooths` *evaluate smooths at new data*

Description

Evaluate a set of smooths at new data locations

Usage

```
evaluate_smooths(x, newdata)
```

Arguments

`x` a greta array created with greta.gam::smooths

`newdata` a dataframe with the same column names and datatypes as that used to create x, with data at which to evauate the smooths

Author(s)

Nick Golding

`greta.gam` *greta.gam: generalised additive models in greta using mgcv*

Description

greta.gam is a module for greta that lets you use mgcv's smoother functions and formula syntax to define smooth terms for use in a greta model. You can then define your own likelihood to complete the model, and fit it by MCMC.

jagam2greta *Turn a jagam model definition into a greta model definition*

Description

Takes a GAM defined by `formula` and returns the corresponding `greta` model via the power of `jagam`. Response variable is generated from dummy data and not used.

Usage

```
jagam2greta(formula, data, newdata, sp = NULL, knots = NULL, tol = 0)
```

Arguments

`formula` a GAM formula representing the smooth terms, as in `mgcv::gam`. Only the right hand side fo the formula will be used.

`data` a data frame or list containing the covariates required by the formula. These covariates cannot be greta arrays.

`sp` an optional vector of smoothing parameters, two per smooth term in the model, in the same order as the formula. If `sp = NULL`, all smoothing parameters will be learned, otherwise all smoothing parameters must be specified by the user. The smoothing parameters may either be a numeric vector or a greta array (which could be a variable).

`knots` an optional list containing user specified knot values to be used for basis construction, as in `mgcv::gam` These knots cannot be greta arrays.

`tol` a non-negative scalar numerical tolerance parameter. You can try increasing this if the model has numerical stability issues

Value

a `list` with the following elements: `betas` a greta array for the coefficients to be estimated (with appropriate priors applied), `X` design matrix for this model, `X_pred` prediction matrix.

smooths *greta array representations of mgcv smooth terms*

Description

`smooths` translates the right hand side of a mgcv GAM formula into a corresponding Bayesian representation of smooth terms. This formula may include multiple combined smooths of different types, as well as fixed effect terms and intercepts. The resulting greta array representing the combined smooth can then be used in a greta model.

Usage

```
smooths(formula, data = list(), knots = NULL, sp = NULL, tol = 0)
```

Arguments

`formula` a GAM formula representing the smooth terms, as in `mgcv::gam`. Only the right hand side fo the formula will be used.

`data` a data frame or list containing the covariates required by the formula. These covariates cannot be greta arrays.

`knots` an optional list containing user specified knot values to be used for basis construction, as in `mgcv::gam` These knots cannot be greta arrays.

`sp` an optional vector of smoothing parameters, two per smooth term in the model, in the same order as the formula. If `sp = NULL`, all smoothing parameters will be learned, otherwise all smoothing parameters must be specified by the user. The smoothing parameters may either be a numeric vector or a greta array (which could be a variable).

`tol` a non-negative scalar numerical tolerance parameter. You can try increasing this if the model has numerical stability issues

Details

Only the right hand side of `formula` will be used to define the smooth terms. The user must complete the gam model by specifying the link and likelihood term in greta. A warning will be issued if the formula has a left hand side.

Note that by default, GAM formulas add an intercept term. If you have already specified an intercept for your greta model, you can remove the intercept from the smooth term by adding `-1` as a term in your formula.

Like `mgcv::jagam`, `smooths` translates a mgcv GAM formula into a Bayesian representation of the smooth terms, using the GAM smoothing penalty matrix as a multivariate normal prior to penalise model fitting. Unlike `gam`, `smooths` does not perform the integration required to penalise model fitting. The model must be fitted by MCMC to carry out this integration - it does not make sense to do maximum likelihood optimisation on a greta model that uses `smooths`.

Examples

```
## Not run:
n <- 30
x <- runif(n, 0, 10)
f <- function(x) {
  sin(x * 2) + 1.6 * (x < 3) - 1.4 * (x > 7)
}
y <- f(x) + rnorm(n, 0, 0.3)
x_plot <- seq(0, 10, length.out = 200)

z <- smooths(~s(x), data = data.frame(x = x))

distribution(y) <- normal(z, 0.3)
```

```
z_pred <- evaluate_smooths(z, newdata = data.frame(x = x_plot))

# build model
m <- model(z_pred)
draws <- mcmc(m, n_samples = 100)

plot(x, y, pch = 19, cex = 0.4, col = "red")
apply(draws[[1]], 1, lines, x = x_plot, col = "blue")
points(x, y, pch = 19, cex = 0.4, col = "red")

## End(Not run)
```

Index