Package 'vinereg'

August 10, 2018

Type Package

Title D-Vine Quantile Regression

Version 0.5.0

Maintainer Thomas Nagler <mail@tnagler.com>

Description Implements D-vine quantile regression models with parametric or nonparametric pair-copulas. See Kraus and Czado (2017) <doi:10.1016/j.csda.2016.12.009> and Schallhorn et al. (2017) <arXiv:1705.08310>.

License GPL-3

LazyData TRUE

Imports cctools, rvinecopulib (>= 0.3.0), future, furrr, kde1d (>= 0.2.0),

RoxygenNote 6.1.0

NeedsCompilation no

Suggests knitr, rmarkdown, ggplot2, PivotalR, quantreg, tidyr, dplyr, purrr, scales, mgcv, testthat, covr

VignetteBuilder knitr

URL https://tnagler.github.io/vinereg

BugReports https://github.com/tnagler/vinereg/issues

Author Thomas Nagler [aut, cre], Dani Kraus [ctb]

Repository CRAN

Date/Publication 2018-08-09 23:00:03 UTC

R topics documented:

plot_effects 2
predict.vinereg 2
vinereg 3

Index **6**

plot_effects *Plot marginal effects of a D-vine regression model*

Description

The marginal effects of a variable is the expected effect, where expectation is meant with respect to all other variables.

Usage

```
plot_effects(object, alpha = c(0.1, 0.5, 0.9), vars = object$order)
```

Arguments

object	a vinereg object
alpha	vector of quantile levels.
vars	vector of variable names.

Examples

```
# simulate data
x <- matrix(rnorm(300), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)
plot_effects(fit)
```

predict.vinereg *Predict conditional mean and quantiles from a D-vine regression model*

Description

Predict conditional mean and quantiles from a D-vine regression model

Usage

```
## S3 method for class 'vinereg'
predict(object, newdata, alpha = 0.5, uscale = FALSE,
  ...)

## S3 method for class 'vinereg'
fitted(object, alpha = 0.5, ...)
```

Arguments

object	an object of class vinereg.
newdata	matrix of covariate values for which to predict the quantile.
alpha	vector of quantile levels; NA predicts the mean based on an average of the 1:10 / 11-quantiles.
uscale	if TRUE input (newdata) and output is on copula scale.
...	unused.

Value

A data.frame of quantiles where each column corresponds to one value of alpha.

See Also

vinereg

Examples

```
# simulate data
x <- matrix(rnorm(300), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
(fit <- vinereg(y ~ ., dat))

# inspect model
summary(fit)
plot_effects(fit)

# model predictions
mu_hat  <- predict(fit, newdata = dat, alpha = NA)          # mean
med_hat <- predict(fit, newdata = dat, alpha = 0.5)         # median

# observed vs predicted
plot(cbind(y, mu_hat))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z.1")))
```

---

vinereg *D-vine regression models*

Description

Sequential estimation of a regression D-vine for the purpose of quantile prediction as described in Kraus and Czado (2017). If discrete variables are declared as `ordered()` or `factor()`, jittering is used to make them continuous (see `cctools::cont_conv()`). Although this may make the model estimate inconsistent, predictions are usually still reasonable.

Usage

```
vinereg(formula, data, family_set = "parametric", selcrit = "loglik",
  order = NA, par_1d = list(), cores = 1, uscale = FALSE, ...)
```

Arguments

`formula`	an object of class "formula"; same as `lm()`.
`data`	data frame (or object coercible by `as.data.frame()`) containing the variables in the model.
`family_set`	see `family_set` argument of `rvinecopulib::bicop()`.
`selcrit`	selection criterion based on conditional log-likelihood. `"loglik"` (default) imposes no correction; other choices are `"aic"` and `"bic"`.
`order`	the order of covariates in the D-vine, provided as vector of variable names (after calling `cctools::cont_conv()` on the `model.frame()`); selected automatically if `order = NA` (default).
`par_1d`	list of options passed to `kde1d::kde1d()`, must be one value for each margin, e.g. `list(xmin = c(0, 0, NaN))` if the response and first covariate have non-negative support.
`cores`	integer; the number of cores to use for computations.
`uscale`	logical indicating whether the data are already on copula scale (no margins have to be fitted).
`...`	further arguments passed to `rvinecopulib::bicop()`.

Value

An object of class vinereg. It is a list containing the elements

formula the formula used for the fit.

selcrit criterion used for variable selection.

model_frame the data used to fit the regression model.

margins list of marginal models fitted by `kde1d::kde1d()`.

vine an `rvinecopulib::vinecop_dist()` object containing the fitted D-vine.

stats fit statistics such as conditional log-likelihood/AIC/BIC and p-values for each variable's contribution.

order order of the covariates chosen by the variable selection algorithm.

selected_vars indices of selected variables.

Use `predict.vinereg()` to predict conditional quantiles. `summary.vinereg()` shows the contribution of each selected variable with the associated p-value derived from a likelihood ratio test.

References

Kraus and Czado (2017), D-vine copula based quantile regression, Computational Statistics and Data Analysis, 110, 1-18

See Also

predict.vinereg

Examples

```
# simulate data
x <- matrix(rnorm(300), 100, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(100, 2, 0.5)))

# fit vine regression model
(fit <- vinereg(y ~ ., dat))

# inspect model
summary(fit)
plot_effects(fit)

# model predictions
mu_hat  <- predict(fit, newdata = dat, alpha = NA)          # mean
med_hat <- predict(fit, newdata = dat, alpha = 0.5)         # median

# observed vs predicted
plot(cbind(y, mu_hat))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z.1")))
```

Index

as.data.frame(), *4*

fitted.vinereg *(*predict.vinereg*)*, 2

kde1d::kde1d(), *4*

lm(), *4*

plot_effects, 2
predict.vinereg, 2, *5*
predict.vinereg(), *4*

rvinecopulib::bicop(), *4*
rvinecopulib::vinecop_dist(), *4*

vinereg, *3*, 3