

Package: vinereg (via r-universe)

November 5, 2024

Type Package

Title D-Vine Quantile Regression

Version 0.10.0

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](https://doi.org/10.1016/j.csda.2016.12.009)> and Schallhorn et al. (2017) <[arXiv:1705.08310](https://arxiv.org/abs/1705.08310)>.

License GPL-3

Imports rvinecopulib (>= 0.6.1.1.2), kde1d (>= 1.0.2), Rcpp, assertthat

LinkingTo rvinecopulib, RcppEigen, Rcpp, BH, wdm, RcppThread, kde1d

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

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

VignetteBuilder knitr

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

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

Encoding UTF-8

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

RemoteUrl <https://github.com/tnagler/vinereg>

RemoteRef HEAD

RemoteSha beed02ab9096f7017d857479733a33a9ba49ddbc

Contents

cll	2
cpdf	3

cpit	3
plot_effects	4
predict.vinereg	5
vinereg	6
Index	9

c11 *Conditional log-likelihood*

Description

Calculates the conditional log-likelihood of the response given the covariates.

Usage

```
c11(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	matrix of response and covariate values for which to compute the conditional distribution.
cores	integer; the number of cores to use for computations.

Examples

```
# simulate data
x <- matrix(rnorm(200), 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)

c11(fit, dat)
fit$stats$c11
```

cpdf *Conditional PDF*

Description

Calculates the conditional density of the response given the covariates.

Usage

```
cpdf(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	matrix of response and covariate values for which to compute the conditional density
cores	integer; the number of cores to use for computations.

Examples

```
# simulate data
x <- matrix(rnorm(200), 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)

cpdf(fit, dat)
```

cpit *Conditional probability integral transform*

Description

Calculates the conditional distribution of the response given the covariates.

Usage

```
cpit(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	matrix of response and covariate values for which to compute the conditional distribution.
cores	integer; the number of cores to use for computations.

Examples

```
# simulate data
x <- matrix(rnorm(200), 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)

hist(cpit(fit, dat)) # should be approximately uniform
```

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(200), 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	<i>Predict conditional mean and quantiles from a D-vine regression model</i>
-----------------	--

Description

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

Usage

```
## S3 method for class 'vinereg'  
predict(object, newdata, alpha = 0.5, cores = 1, ...)  
  
## 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.
cores	integer; the number of cores to use for computations.
...	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(200), 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).

Usage

```

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

```

Arguments

formula	an object of class "formula"; same as <code>lm()</code> .
data	data frame (or object coercible by <code>as.data.frame()</code>) containing the variables in the model.
family_set	see family_set argument of <code>rvinecopulib::bicop()</code> .
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 <code>vinereg:::expand_factors(model.frame(formula, data))</code>); selected automatically if order = NA (default).
par_1d	list of options passed to <code>kde1d::kde1d()</code> , must be one value for each margin, e.g. <code>list(xmin = c(0, 0, NaN))</code> if the response and first covariate have non-negative support.

<code>weights</code>	optional vector of weights for each observation.
<code>cores</code>	integer; the number of cores to use for computations.
<code>...</code>	further arguments passed to <code>rvinecopulib::bicop()</code> .
<code>uscale</code>	if TRUE, <code>vinereg</code> assumes that marginal distributions have been taken care of in a preliminary step.

Details

If discrete variables are declared as `ordered()` or `factor()`, they are handled as described in Panagiotelis et al. (2012). This is different from previous version where the data was jittered before fitting.

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

Panagiotelis, A., Czado, C., & Joe, H. (2012). Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499), 1063-1072.

See Also

[predict.vinereg](#)

Examples

```
# simulate data
x <- matrix(rnorm(200), 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()`, 6

`c11`, 2

`cpdf`, 3

`cpit`, 3

`fitted.vinereg(predict.vinereg)`, 5

`kde1d::kde1d()`, 6, 7

`lm()`, 6

`plot_effects`, 4

`predict.vinereg`, 5, 7

`predict.vinereg()`, 7

`rvinecopulib::bicop()`, 6, 7

`rvinecopulib::vinecop_dist()`, 7

`vinereg`, 5, 6