

Package: MLCOPULA (via r-universe)

September 5, 2024

Type Package

Title Classification Models with Copula Functions

Description Provides several classifiers based on probabilistic models. These classifiers allow to model the dependence structure of continuous features through bivariate copula functions and graphical models, see Salinas-Gutiérrez et al. (2014) <[doi:10.1007/s00180-013-0457-y](https://doi.org/10.1007/s00180-013-0457-y)>.

Version 1.0.0

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.1

Imports copula, igraph, GRIDCOPULA, kde1d, pracma, TSP

Author Rogelio Salinas Gutiérrez [aut, cre, cph] (<<https://orcid.org/0000-0002-1669-4460>>), Angélica Hernández Quintero [aut, cph] (<<https://orcid.org/0009-0000-0400-1244>>), Pedro Abraham Montoya Calzada [aut, cph] (<<https://orcid.org/0009-0002-3497-210X>>), Carlos Alberto López Hernández [aut, cph] (<<https://orcid.org/0009-0000-9428-1293>>), Juan Manuel Marquez Romero [aut, cph] (<<https://orcid.org/0000-0002-5059-1174>>)

Maintainer Rogelio Salinas Gutiérrez <rsalinas@correo.uaa.mx>

NeedsCompilation no

Repository CRAN

Date/Publication 2024-06-06 17:00:05 UTC

Contents

copulaClassifier	2
copulaPredict	3
Index	5

copulaClassifier *Train a classification model using copula functions.*

Description

It trains a classification model based on copulas, the joint density of copulas is built with tree or chain graphic model, as shown in Salinas-Gutiérrez et al., 2014.

Usage

```
copulaClassifier(
  X,
  y,
  distribution = "kernel",
  copula = "frank",
  weights = "likelihood",
  graph_model = "tree",
  k = 7,
  m = 7,
  method_grid = "ml"
)
```

Arguments

X	Data frame with n samples and $d > 1$ predictor variables.
y	a vector of size n , with the classes to predict.
distribution	Marginal distribution to be used: "normal" or "kernel", by default kernel.
copula	Character or vector with the name of the copula to be used: "frank", "gaussian", "clayton", "joe", "gumbel", "amh", "grid", by default "frank". For parametric copulas, "frank", "gaussian", "clayton", "joe", "gumbel", "amh", one or more copulas can be selected. For nonparametric copula, only "grid" can be selected. See the examples for more details.
weights	Character with the weight construction method: "likelihood" or "mutual_information", by default "likelihood".
graph_model	Character with the graphical model structure: "tree" or "chain", by default "tree".
k	Only for grid copula. Positive integer indicating the number of subintervals for the U_2 variable.
m	Only for grid copula. Positive integer indicating the number of subintervals for the U_1 variable.
method_grid	Only for grid copula. Fitting method, least squares "ls" or maximum likelihood "ml", by default "ml".

Value

Returns a list with the trained model.

References

Salinas-Gutiérrez, R., Hernández-Aguirre, A., Villa-Diharce, E.R. (2014). Copula selection for graphical models in continuous Estimation of Distribution Algorithms. *Computational Statistics*, **29**(3–4):685–713. doi:10.1007/s001800130457y

Examples

```
X <- iris[,1:4]
y <- iris$Species
model <- copulaClassifier(X = X, y = y, copula = "frank",
                        distribution = "kernel", graph_model = "tree")
y_pred <- copulaPredict(X = X, model = model)
table(y,y_pred$class)
#Example 2
X <- iris[,1:4]
y <- iris$Species
model <- copulaClassifier(X = X, y = y, copula = c("frank","clayton"),
                        distribution = "kernel", graph_model = "chain")
y_pred <- copulaPredict(X = X, model = model)
table(y,y_pred$class)
```

copulaPredict

Get predictions from a classification model.

Description

Use the models trained with copula functions to generate new predictions.

Usage

```
copulaPredict(X, model)
```

Arguments

X	Data frame with predictor variables.
model	Classification model.

Value

A list with the prediction of the class "class" and the probabilities of each class "prob".

Examples

```
X <- iris[,1:4]
y <- iris$Species
model <- copulaClassifier(X = X, y = y, copula = "frank",
                        distribution = "kernel", graph_model = "tree")
y_pred <- copulaPredict(X = X, model = model)
```

```
table(y,y_pred$class)
#Example 2
X <- iris[,1:4]
y <- iris$Species
model <- copulaClassifier(X = X, y = y, copula = c("frank","clayton"),
                        distribution = "kernel", graph_model = "chain")
y_pred <- copulaPredict(X = X, model = model)
table(y,y_pred$class)
```

Index

`copulaClassifier`, 2
`copulaPredict`, 3