

# Package ‘MLCOPULA’

January 20, 2025

**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** 1.0.1

**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](mailto:rsalinas@correo.uaa.mx)>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-10-23 05:10:01 UTC

## Contents

classification_report . . . . .	2
copulaClassifier . . . . .	3
copulaPredict . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

classification\_report *Calculates classification performance metrics.*

---

### Description

Calculates the confusion matrix and several performance metrics.

### Usage

```
classification_report(  
  y_true,  
  y_pred  
)
```

### Arguments

y_true	A vector with the true labels.
y_pred	A vector with the predicted labels.

### Value

Returns a list with the following entries:

metrics	A table with the precision, recall and f1-score for each class.
confusion_matrix	The confusion matrix.
accuracy	The accuracy.
mutual_information	The mutual information between the true and the predicted classes.

### Examples

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

---

copulaClassifier      *Trains a classification model using copula functions.*

---

### Description

It trains a classification model based on copulas. The dependence structure of the joint density is built by using a graphical model along with bivariate copulas, 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	Either a character or a string vector with the name of the copula to be used: "amh", "clayton", "frank", "gaussian", "grid", "gumbel", "independent" and "joe", by default "frank". For parametric copulas, "amh", "clayton", "frank", "gaussian", "gumbel", and "joe", one or more copulas can be selected. For nonparametric copula, only "grid" can be selected. See the examples for more details.
weights	A character with the weight construction method for the graphical model: "likelihood" or "mutual_information", by default "likelihood".
graph_model	A character with the graphical model structure: "tree" or "chain", by default "tree".
k	Only for the grid copula. Positive integer indicating the number of subintervals for the $U_2$ variable.
m	Only for the grid copula. Positive integer indicating the number of subintervals for the $U_1$ variable.
method_grid	Only for the grid copula. Fitting method, least squares "ls" or maximum likelihood "ml", by default "ml".

**Value**

Returns a 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**

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

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

---

copulaPredict

*Gets 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	A classification model given by <code>copulaClassifier</code> .

**Value**

A list with the following entries:

class	a vector with the predicted class.
prob	a data frame with the probabilities of each class.

**Examples**

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

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

# Index

classification\_report, 2  
copulaClassifier, 3, 4  
copulaPredict, 4