Package ‘MMDCopula’

October 13, 2020

Type Package
Title Robust Estimation of Copulas by Maximum Mean Discrepancy
Version 0.1.0
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Depends R (>= 3.6.0)
Imports VineCopula, cubature, pcaPP, randtoolbox,
Suggests knitr, rmarkdown
VignetteBuilder knitr
BugReports https://github.com/AlexisDerumigny/MMDCopula/issues
NeedsCompilation no
Author Alexis Derumigny [aut, cre] (<https://orcid.org/0000-0002-6163-8097>), Pierre Alquier [aut] (<https://orcid.org/0000-0003-4249-7337>), Jean-David Fermanian [aut] (<https://orcid.org/0000-0001-5960-5555>), Badr-Eddine Chérief-Abdellatif [aut]
Maintainer Alexis Derumigny <a.f.f.derumigny@utwente.nl>
Repository CRAN
Date/Publication 2020-10-13 14:00:06 UTC

R topics documented:

BiCopEst.MO ............................................................... 2
BiCopEstMMD ............................................................. 3
BiCopGradMMD .......................................................... 6
BiCopParamDistLp ....................................................... 8
BiCopSim.MO ............................................................. 10
BiCopEst.MO

Estimation of Marshall-Olkin copulas

Description

Estimation of Marshall-Olkin copulas

Usage

BiCopEst.MO(
  u1,
  u2,
  method,
  par.start = 0.5,
  kernel = "gaussian.KG",
  gamma = 0.95,
  alpha = 1,
  niter = 100,
  ndrawings = 10,
  naveraging = 1
)

Arguments

u1 vector of observations of the first coordinate, in [0, 1].
u2 vector of observations of the second coordinate, in [0, 1].
method a character giving the name of the estimation method, among:
  • curve: $\alpha$ is estimated by inversion of the probability measure of the diagonal \{(u, v) : u = v\}
  • itau: $\alpha$ is estimated by inversion of Kendall’s tau
  • MMD: $\alpha$ is estimated by MMD optimization
par.start starting parameter of the gradient descent. (only used for method = "MMD")
kernelpar the kernel used in the MMD distance (only used for method = "MMD") : it can be a function taking in parameter (u1, u2, v1, v2, gamma, alpha) or a name giving the kernel to use in the list:
  • gaussian: Gaussian kernel $k(x, y) = \exp(-\|x-y\|_2^2)$
  • exp.12: $k(x, y) = \exp(-\|x-y\|_2)$
  • exp.11: $k(x, y) = \exp(-\|x-y\|_1)$
  • inv.12: $k(x, y) = 1/(1 + \|x-y\|_2^\alpha)$
  • inv.11: $k(x, y) = 1/(1 + \|x-y\|_1^\alpha)$

Each of these names can receive the suffix ".KG", such as "gaussian.KG" to indicates that the kernel $k(x, y)$ is replaced by $k(\Phi^{-1}(x), \Phi^{-1}(y))$ where $\Phi^{-1}$ denotes the quantile function of the standard Normal distribution.
BiCopEstMMD

<table>
<thead>
<tr>
<th>gamma</th>
<th>parameter $\gamma$ to be used in the kernel. (only used for method = &quot;MMD&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>parameter $\alpha$ to be used in the kernel, if any. (only used for method = &quot;MMD&quot;)</td>
</tr>
<tr>
<td>niter</td>
<td>number of iterations of the stochastic gradient algorithm. (only used for method = &quot;MMD&quot;)</td>
</tr>
<tr>
<td>ndrawings</td>
<td>number of replicas of the stochastic estimate of the gradient drawn at each step. The gradient is computed using the average of these replicas. (only used for method = &quot;MMD&quot;)</td>
</tr>
<tr>
<td>naveraging</td>
<td>number of full run of the stochastic gradient algorithm that are averaged at the end to give the final estimated parameter. (only used for method = &quot;MMD&quot;)</td>
</tr>
</tbody>
</table>

References


See Also

BiCopSim_MO for the estimation of Marshall-Olkin copulas. BiCopEstMMD for the estimation of other parametric copula families by MMD.

Examples

```r
U <- BiCopSim_MO(n = 1000, alpha = 0.2)
estimatedPar <- BiCopEst.MO(u1 = U[,1], u2 = U[,2], method = "MMD", niter = 1, ndrawings = 1)
estimatedPar <- BiCopEst.MO(u1 = U[,1], u2 = U[,2], method = "MMD")
```

BiCopEstMMD

Estimation of parametric bivariate copulas using stochastic gradient descent on the MMD criteria

Description

This function uses computes the MMD-estimator of a bivariate copula family. This computation is done through a stochastic gradient algorithm, that is itself computed by the function BiCopGradMMD(). The main arguments are the two vectors of observations, and the copula family. The bidimensional copula families are indexed in the same way as in VineCopula::BiCop() (which computes the MLE estimator).
BiCopEstMMD

Usage

\texttt{BiCopEstMMD(}
  \texttt{u1,}
  \texttt{u2,}
  \texttt{family,}
  \texttt{tau = 0.1,}
  \texttt{par = NULL,}
  \texttt{par2 = NULL,}
  \texttt{kernel = "gaussian",}
  \texttt{gamma = 0.23,}
  \texttt{alpha = 1,}
  \texttt{niter = 100,}
  \texttt{epsilon = 1e-04,}
  \texttt{method = "QMCV",}
  \texttt{quasiRNG = "sobol",}
  \texttt{ndrawings = 10})}

Arguments

\texttt{u1} vector of observations of the first coordinate, in \([0, 1]\).
\texttt{u2} vector of observations of the second coordinate, in \([0, 1]\).
\texttt{family} the chosen family of copulas (see the documentation of the class \texttt{VineCopula::BiCop()} for the available families).
\texttt{tau} the copula family can be parametrized by the parameter \texttt{par} or by Kendall’s \texttt{tau}. Here, the user can choose the initial value of \texttt{tau} for the stochastic gradient algorithm.
\texttt{par} if different from \texttt{NULL}, the parameter \texttt{tau} is ignored, and the initial parameter must be given here. The initial Kendall’s \texttt{tau} is then computed thanks to \texttt{VineCopula::BiCopPar2Tau()}.
\texttt{par2} initial value for the second parameter, if any. (Works only for Student copula).
\texttt{kernel} the kernel used in the MMD distance: it can be a function taking in parameter \((u1,u2,v1,v2,gamma,\alpha)\) or a name giving the kernel to use in the list:
  \begin{itemize}
  \item \texttt{gaussian}: Gaussian kernel \(k(x, y) = \exp(-\|x-y\|^2_2)\)
  \item \texttt{exp.12}: \(k(x, y) = \exp(-\|x-y\|_2)\)
  \item \texttt{exp.11}: \(k(x, y) = \exp(-\|x-y\|_1)\)
  \item \texttt{inv.12}: \(k(x, y) = 1/(1 + \|x-y\|_2)^\alpha\)
  \item \texttt{inv.11}: \(k(x, y) = 1/(1 + \|x-y\|_1)^\alpha\)
  \end{itemize}
Each of these names can receive the suffix ".KG", such as "gaussian.KG" to indicates that the kernel \(k(x, y)\) is replaced by \(k(\Phi^{-1}(x), \Phi^{-1}(y))\) where \(\Phi^{-1}\) denotes the quantile function of the standard Normal distribution.
\texttt{gamma} parameter \(\gamma\) to be used in the kernel.
\texttt{alpha} parameter \(\alpha\) to be used in the kernel, if any.
BiCopEstMMD

niter number of iterations of the stochastic gradient algorithm.

epsilon the differential of VineCopula::BiCopTau2Par() is computed thanks to a finite difference with increment epsilon.

method the method of computing the stochastic gradient:

• MC: classical Monte-Carlo with ndrawings replications.
• QMCV: usual Monte-Carlo on U with ndrawings replications, quasi Monte-Carlo on V.

quasiRNG a function giving the quasi-random points in \([0, 1]^2\) or a name giving the method to use in the list:

• sobol: use of the Sobol sequence implemented in randtoolbox::sobol
• halton: use of the Halton sequence implemented in randtoolbox::halton
• torus: use of the Torus sequence implemented in randtoolbox::torus

ndrawings number of replicas of the stochastic estimate of the gradient drawn at each step. The gradient is computed using the average of these replicas.

Value

an object of class VineCopula::BiCop() containing the estimated copula.

References


See Also

VineCopula::BiCopEst() for other methods of estimation such as Maximum Likelihood Estimation or Inversion of Kendall’s tau. BiCopGradMMD() for the computation of the stochastic gradient. BiCopEst.MO for the estimation of Marshall-Olkin copulas by MMD.

Examples

# Estimation of a bivariate Gaussian copula with correlation 0.5.
dataSampled = VineCopula::BiCopSim(N = 500, family = 1, par = 0.5)
estimator = BiCopEstMMD(u1 = dataSampled[,1], u2 = dataSampled[,2], family = 1, niter=10)
estimator$par

# Estimation of a bivariate Student copula with correlation 0.5 and 5 degrees of freedom
dataSampled = VineCopula::BiCopSim(N = 1000, family = 2, par = 0.5, par2 = 5)
estimator = BiCopEstMMD(u1 = dataSampled[,1], u2 = dataSampled[,2], family = 2)
estimator$par

# Comparison with maximum likelihood estimation with and without outliers
dataSampled = VineCopula::BiCopSim(N = 500, family = 1, par = 0.5)
estimatorMMD = BiCopEstMMD(u1 = dataSampled[,1], u2 = dataSampled[,2], family = 1)
BiCopGradMMD

Computation of the gradient of the MMD criterion for parametric bi-variate copulas models

Description

This function computes a stochastic estimate of the gradient of the MMD criterion for parametric estimation of bidimensional copula family. The main arguments are the two vectors of observations, and the copula family. The family is parametrized as in VineCopula::BiCop(), using the Kendall’s tau instead of the first parameter. This function is used by BiCopEstMMD() to perform parameter estimation via MMD minimization.

Usage

BiCopGradMMD(
  u1,
  u2,
  family,
  tau,
  par = NULL,
  par2 = 0,
  kernel = "gaussian.KG",
  gamma = 0.95,
  alpha = 1,
  epsilon = 1e-04,
  method = "QMCV",
  quasiRNG = "sobol",
)

estimatorMMD$par
estimatorMLE = VineCopula::BiCopEst(u1 = dataSampled[,1], u2 = dataSampled[,2],
  family = 1, method = "mle")
estimatorMLE$par
dataSampled[1:10,1] = 0.999
dataSampled[1:10,2] = 0.001
estimatorMMD = BiCopEstMMD(u1 = dataSampled[,1], u2 = dataSampled[,2], family = 1)
estimatorMMD$par
estimatorMLE = VineCopula::BiCopEst(u1 = dataSampled[,1], u2 = dataSampled[,2],
  family = 1, method = "mle")
estimatorMLE$par

# Estimation of a bivariate Gaussian copula with real data
data("daxreturns", package = "VineCopula")
BiCopEstMMD(u1 = daxreturns[,1], u2 = daxreturns[,2], family = 1)
estimator$par
BiCopGradMMD

    ndrawings = 10

Arguments

u1    vector of observations of the first coordinate, in [0, 1].
u2    vector of observations of the second coordinate, in [0, 1].
family the chosen family of copulas (see the documentation of the class VineCopula::BiCop() for the available families).
tau the copula family can be parametrized by the parameter par or by Kendall’s tau. This function assumes a Kendall tau parametrization. Thus, the user can choose the value of Kendall tau at which the stochastic gradient should be computed.
par    if different from NULL, the user must instead of tau specify the corresponding parameter par. The value of tau is then ignored.
par2    value for the second parameter, if any. (Works only for Student copula).
kernel the kernel used in the MMD distance: it can be a function taking in parameter (u1,u2,v1,v2,gamma, alpha) or a name giving the kernel to use in the list:
• gaussian: Gaussian kernel \( k(x, y) = \exp(-\|x-y\|_2^2) \)
• exp.12: \( k(x, y) = \exp(-\|x-y\|_2) \)
• exp.11: \( k(x, y) = \exp(-\|x-y\|_1) \)
• inv.12: \( k(x, y) = 1/(1 + \|x-y\|_2^a) \)
• inv.11: \( k(x, y) = 1/(1 + \|x-y\|_1^a) \)

Each of these names can receive the suffix "._KG", such as "gaussian._KG" to indicates that the kernel \( k(x, y) \) is replaced by \( k(\Phi^{-1}(x), \Phi^{-1}(y)) \) where \( \Phi^{-1} \) denotes the quantile function of the standard Normal distribution.
gamma parameter \( \gamma \) to be used in the kernel.
alpha parameter \( \alpha \) to be used in the kernel, if any.
epsilon the differential of VineCopula::BiCopTau2Par() is computed thanks to a finite difference with increment epsilon.
method the method of computing the stochastic gradient:
• MC: classical Monte-Carlo with ndrawings replications.
• QMCV: usual Monte-Carlo on U with ndrawings replications, quasi Monte-Carlo on V.
quasiRNG a function giving the quasi-random points in \([0, 1]^2\) or a name giving the method to use in the list:
• sobol: use of the Sobol sequence implemented in randtoolbox::sobol
• halton: use of the Halton sequence implemented in randtoolbox::halton
• torus: use of the Torus sequence implemented in randtoolbox::torus
ndrawings number of replicas of the stochastic estimate of the gradient drawn at each step. The gradient is computed using the average of these replicas.
Value

the value of the gradient.

References


See Also

BiCopEstMMD() for the estimation of parametric bivariate copulas by stochastic gradient descent on the MMD criteria.

Examples

# Simulation from a bivariate Gaussian copula with correlation 0.5.
dataSampled = VineCopula::BiCopSim(N = 500, family = 1, par = 0.5)

# computation of the gradient of the MMD criteria at different points
# Gradient is small at the true parameter
BiCopGradMMD(dataSampled[,1], dataSampled[,2], family = 1, par = 0.5)
# Gradient is negative when below the parameter
BiCopGradMMD(dataSampled[,1], dataSampled[,2], family = 1, par = 0.1)
# and positive when above
BiCopGradMMD(dataSampled[,1], dataSampled[,2], family = 1, par = 0.8)

BiCopParamDistLp

Compute the distance between 2 parametric copulas

Description

This function uses the numerical integration procedure cubature::hcubature() to numerical integrate the distance between the distribution or between the densities of two bivariate copulas.

Usage

BiCopParamDistLp(
  family,
  par,
  par_p,
  par2 = par,
  par2_p = par_p,
  family_p = family,
  p,
  type,
  maxEval = 0
)
Arguments

- **family**: family of the first copula.
- **par**: first parameter of the first copula.
- **par_p**: first parameter of the second copula.
- **par2**: second parameter of the first copula (only useful for two-parameter families of copulas).
- **par2_p**: second parameter of the first copula (only useful for two-parameter families of copulas).
- **family_p**: family of the second copula.
- **p**: determines the $L_p$ distance that is used.
- **type**: type of the functions considered. Can be `cdf` for the distance between the two cumulative distribution functions or `pdf` for the distance between the two probability density functions.
- **maxEval**: maximum number of evaluation of the function be integrated. If 0, then no maximum limit is given.

Value

A list of four items:

- **distance**: the value of the distance
- **integral**: the value of the integral, which is the $p$-th power of the distance.
- **error**: the estimated relative error of the integral
- **returnCode**: the integer return code of the C routine called by `cubature::hcubature()`. This should be 0 if there is no error.

Examples

```r
# Distance between the densities of a Gaussian copula with correlation 0.5
# and a Gaussian copula with correlation 0.2
BiCopParamDistLp(family = 1, par = 0.5, par_p = 0.2, p = 2, type = "cdf", maxEval = 10)

# Distance between the cdf of a Student copula
# with correlation 0.5 and 4 degrees of freedom
# and a Student copula with the same correlation but 20 degrees of freedom
BiCopParamDistLp(family = 2, par = 0.5, par_p = 0.5, par2 = 5, par2_p = 20, p = 2, type = "pdf", maxEval = 10)

# Distance between the densities of a Gaussian copula with correlation 0.5
# and of a Student copula with correlation 0.5 and 15 degrees of freedom
BiCopParamDistLp(family = 1, par = 0.5, par_p = 0.5, par2_p = 15, family_p = 2, p = 2, type = "pdf", maxEval = 10)
```
BiCopSim.MO

Simulation of Marshall-Olkin copula

Description

This functions simulates independent realizations from the Marshall-Olkin copula.

Usage

BiCopSim.MO(n, alpha)

Arguments

n    number of samples
alpha parameter of the Marshall-Olkin copula

Value

an $n \times 2$ matrix containing the samples

See Also

BiCopEst.MO for the estimation of Marshall-Olkin copulas.

Examples

# Simulation from a Marshall-Olkin copula with parameter alpha = 0.5
BiCopSim.MO(n = 100, alpha = 0.5)
Index

BiCop, 3–7
BiCopEst, 5
BiCopEst.MO, 2, 5, 10
BiCopEstMMD, 3, 3, 6, 8
BiCopGradMMD, 3, 5, 6
BiCopPar2Tau, 4
BiCopParamDistLp, 8
BiCopSim.MO, 3, 10
BiCopTau2Par, 5, 7

halton, 5, 7
hcubature, 8, 9

sobol, 5, 7

torus, 5, 7