Inference on copula models

H. A. Mohtashami-Borzadaran N. Doodman M. Amini

Ferdowsi University of Mashhad

Webpage: https://hamb8066.github.io/IntroCopula

Outline

In this part, we are going to do inference on the selected copula model, which consists of

- Parameter estimation
- Parameter estimation based on full model
- Goodness of fit tests
- Model selection
- Non-parametric methods

Parameter estimation

- Maximum likelihood
- Method of moments: Rho inversion
- Method of moments: Tau inversion

Parameter estimation: Maximum likelihood

MLE of copula with parametric margins

```r
library(copula)
mydist <- mvdc(claytonCopula(param = 1.48, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
u <- pgamma(fake.data[,1],shape=2,rate=3)
v <- pgamma(fake.data[,2],shape=4,rate=5)
U<-cbind(u,v)
fitCopula(claytonCopula(),data=U,method="ml")
```

```
## Call: fitCopula(copula, data = data, method = "ml")
## Fit based on "maximum likelihood" and 500 2-dimensional observations.
## Copula: claytonCopula
## alpha
##  1.57
## The maximized loglikelihood is 177
## Optimization converged
```

Parameter estimation: Maximum likelihood

MLE of copula with non-parametric margins

First, the marginal distributions are estimated based on empirical distribution

$$F_{n,j}(x) = \frac{1}{n+1} \sum_{i=1}^{n} I(X_{ij} \leq x), \quad x \in R.$$

The estimated margins are then typically used to form the sample

$$U_{i,n} = (F_{n,1}(X_{i1}), ..., F_{n,d}(X_{id})), \quad i \in \{1, ..., n\},$$

If the margins $F_1, ..., F_d$ were known, the maximum likelihood estimator of θ would be

$$\theta = \arg \sup_{\theta \in \Theta} \sum_{i=1}^{n} \log \left(c_\theta(u_{i1}, ..., u_{id}) \right).$$

Parameter estimation: Maximum likelihood

MLE of copula with non-parametric margins (Copula package)

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
U<-pobs(fake.data)
fitCopula(gumbelCopula(),data=U,method="mpl")
```

```
## Call: fitCopula(copula, data = data, method = "mpl")
## Fit based on "maximum pseudo-likelihood" and 500 2-dimensional observations.
## Copula: gumbelCopula
## alpha
## 1.945
## The maximized loglikelihood is 172.4
## Optimization converged
```

Parameter estimation: Maximum likelihood

MLE of copula with non-parametric margins (Vine-copula package)

```
library(copula)
library(VineCopula)
```

```
##
## Attaching package: 'VineCopula'

## The following object is masked from 'package:copula':
##
##     pobs
```

```
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
U<-pobs(fake.data)
BiCopEst(U[,1],U[,2],family=4,method = "mle")
```

```
## Bivariate copula: Gumbel (par = 2.18, tau = 0.54)
```

Parameter estimation: Rho inversion

First the empirical Spearman's rho ρ_n is calculated and then θ is solved from the following eqation

$$\rho_n = 12 \int \int_{I^2} \Big[C_\theta(u, v) - uv \Big] dudv.$$

```
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
U<-pobs(fake.data)
fitCopula(gumbelCopula(),data=U,method="irho")
```

```
## Call: fitCopula(copula, data = data, method = "irho")
## Fit based on "inversion of Spearman's rho" and 500 2-dimensional observations.
## Copula: gumbelCopula
## alpha
## 1.945
```

This method is only in "copula package"!

Parameter estimation: Tau inversion

Let the marginal parameters be estimated usingFirst the empirical Kendall's Tau τ_n is calculated and then θ is solved from the following eqation

$$\tau_n = 4 \int \int_{I^2} C_\theta(u, v) dC_\theta(u, v) - 1.$$

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
U<-pobs(fake.data)
fitCopula(gumbelCopula(),data=U,method="itau")
```

```
## Call: fitCopula(copula, data = data, method = "itau")
## Fit based on "inversion of Kendall's tau" and 500 2-dimensional observations.
## Copula: gumbelCopula
## alpha
## 2.022
```

Parameter estimation: Tau inversion

```r
library(copula)
library(VineCopula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(1000, mydist)
U<-pobs(fake.data)
BiCopEst(U[,1],U[,2],family=4,method = "itau")
```

```
## Bivariate copula: Gumbel (par = 1.98, tau = 0.49)
```

Parameter estimation based on full model

In this method, the parametric margins and the corresponding parametric copula is selected and then all of the parameters (marginal $\gamma_1, ..., \gamma_d$ and dependence θ parameters) are estimated at once. The log-likelihood would be

$$
\begin{aligned}
l_n(\gamma_1, ..., \gamma_d, \theta) &= \sum_{i=1}^{n} \log c_\theta\Big(F_{1,\gamma_1}(X_{i1}), ..., F_{d,\gamma_d}(X_{id})\Big) \\
&\quad + \sum_{j=1}^{d} \sum_{i=1}^{n} \log f_{j,\gamma_j}(X_{ij}).
\end{aligned}
$$

Parameter estimation based on full model

Only in “Copula” package!

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(500, mydist)
mle1 <- fitMvdc(fake.data, mydist,start = c(2,3,4,5,2))
summary(mle1)
```

```
## Call: fitMvdc(data = fake.data, mvdc = mydist, start = c(2, 3, 4, 5,
##     2))
## Maximum Likelihood estimation based on 500 2-dimensional observations.
## Copula:  gumbelCopula
## Margin 1 :
##          Estimate Std. Error
## m1.shape    2.095      0.120
## m1.rate     3.180      0.203
## Margin 2 :
##          Estimate Std. Error
## m2.shape    4.397      0.261
## m2.rate     5.617      0.352
## Gumbel copula, dim. d = 2
##       Estimate Std. Error
## alpha    1.893      0.083
## The maximized loglikelihood is -242.1
## Optimization converged
## Number of loglikelihood evaluations:
## function gradient
##       42       10
```

Parameter estimation based on full model

The computational advantage of the previous approach generally comes at a price of an efficiency loss relative to the maximum likelihood estimator. If the margins are partially misspecified, the estimation of θ will be biased.

Goodness of fit tests

- Copula package
  - Cramer Von-Mises GOF test
  - Multiplier GOF test
  - GOF test based on method of moments
- Vine-copula package
  - White GOF test
  - Kendall's process GOF test

Goodness of fit tests: Cramer Von-Mises GOF test

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : C \in \zeta \qquad \text{vs} \qquad H_1 : C \notin \zeta.$$

The statistic for this test is

$$S_n^{gof} = \int_{[0,1]^d} n(C_n(u) - C_{\theta_n}(u))^2 dC_n(u),$$

and in empirical version would be

$$S_n^{gof} = \sum_{i=1}^{n} (C_n(U_{i,n}) - C_{\theta_n}(U_{i,n}))^2.$$

For the used algorithm see Hofert et al. (2018) page 182.

Goodness of fit tests: Cramer Von-Mises GOF test

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(50, mydist)
U<-pobs(fake.data)
library(tictoc)
tic()
gofCopula(gumbelCopula(dim=2), x=U, optim.method="BFGS")
```

```
##
##  Parametric bootstrap-based goodness-of-fit test of Gumbel copula, dim.
##  d = 2, with 'method'="Sn", 'estim.method'="mpl":
##
## data:  x
## statistic = 0.029384, parameter = 1.6339, p-value = 0.2223
```

```r
toc()
```

```
## 35.44 sec elapsed
```

Goodness of fit tests: Multiplier GOF test

The main inconvenience of the CV goodness-of-fit test based on the parametric bootstrap is its high computational cost, as each iteration requires both random number generation from the fitted copula and estimation of the copula parameters. In order to circumvent this high computational cost, a faster, large-sample testing procedure based on multiplier central limit theorem.

The test hypothesis

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : C \in \zeta \qquad \text{vs} \qquad H_1 : C \notin \zeta.$$

Goodness of fit tests: Multiplier GOF test

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(50, mydist)
U<-pobs(fake.data)
library(tictoc)
tic()
gofCopula(gumbelCopula(dim = 2), x = U, simulation = "mult")
```

```
##
##  Multiplier bootstrap-based goodness-of-fit test of Gumbel copula, dim.
##  d = 2, with 'method'="Sn", 'estim.method'="mpl":
##
## data:  x
## statistic = 0.022651, parameter = 2.1914, p-value = 0.3082
```

```r
toc()
```

```
## 0.13 sec elapsed
```

Goodness of fit tests: GOF test based on method of moments

It seems that the multiplier goodness-of-fit test based on maximum pseudo-likelihood estimation holds its level reasonably well when applied to small samples. For small sample sizes GOF w.r.t the Kendall's Tau is needed and useful.

The test hypothesis

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : C \in \zeta \qquad \text{vs} \qquad H_1 : C \notin \zeta.$$

Goodness of fit tests: GOF test based on method of moments

```r
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(50, mydist)
U<-pobs(fake.data)
library(tictoc)
tic()
gofCopula(gumbelCopula(dim = 2), x = U, estim.method="itau")
```

```
##
##  Parametric bootstrap-based goodness-of-fit test of Gumbel copula, dim.
##  d = 2, with 'method'="Sn", 'estim.method'="itau":
##
## data:  x
## statistic = 0.027085, parameter = 1.6781, p-value = 0.3412
```

```r
toc()
```

```
## 1.56 sec elapsed
```

Goodness of fit tests: White GOF test

This goodness-of fit test uses the information matrix equality of White (1982) and was investigated by Huang and Prokhorov (2011). The main contribution is that under correct model specification the Fisher Information can be equivalently calculated as minus the expected Hessian matrix or as the expected outer product of the score function.

The test hypothesis

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : \boldsymbol{H}(\theta) + \boldsymbol{C}(\theta) = 0 \qquad \text{vs} \qquad H_0 : \boldsymbol{H}(\theta) + \boldsymbol{C}(\theta) \neq 0,$$

where $\boldsymbol{H}(\theta)$ is the expected Hessian matrix and $\boldsymbol{C}(\theta)$ is the expected outer product of the score function.

Goodness of fit tests: White GOF test

The speed of this test is much more better than the previous tests!

The test hypothesis

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : C \in \zeta \qquad \text{vs} \qquad H_1 : C \notin \zeta.$$

```r
library(copula)
library(VineCopula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(50, mydist)
U<-pobs(fake.data)
BiCopGofTest(U[,1],U[,2],family=4,method = "white")$p.value
```

```
## [1] 1
```

Goodness of fit tests: Kendall's process GOF test

Let ζ be the class of copula model with parameter θ. Based on this test

$$H_0 : C \in \zeta \qquad \text{vs} \qquad H_1 : C \notin \zeta.$$

```
library(copula)
library(VineCopula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(50, mydist)
U<-pobs(fake.data)
BiCopGofTest(U[,1],U[,2],family=4,method = "kendall")
```

```
## $p.value.CvM
## [1] 0.99
##
## $p.value.KS
## [1] 0.94
##
## $statistic.CvM
## [1] 0.02865792
##
## $statistic.KS
## [1] 0.4809452
```

Model selection

In the classical likelihood-based setting of parametric statistics, a famous criteria for model selection is the Akaike information criterion (AIC). In that case, the formula is

$$AIC = 2(l_{n,max} - p),$$

where $l_{n,max}$ is the maximized likelihood and p is the total number of marginal and copula parameters.

Model selection

```
library(copula)
gofCopula(gumbelCopula(dim = 2), x = U, estim.method="mpl")
```

```
##
##  Parametric bootstrap-based goodness-of-fit test of Gumbel copula, dim.
##  d = 2, with 'method'="Sn", 'estim.method'="mpl":
##
## data:  x
## statistic = 0.020386, parameter = 2.5992, p-value = 0.5569
```

```
gofCopula(joeCopula(dim = 2), x = U, estim.method="mpl")
```

```
##
##  Parametric bootstrap-based goodness-of-fit test of Joe copula, dim. d
##  = 2, with 'method'="Sn", 'estim.method'="mpl":
##
## data:  x
## statistic = 0.019999, parameter = 3.6746, p-value = 0.6598
```

```
Log.lik1 <- fitCopula(gumbelCopula(),data=U,method="mpl")@ loglik
Log.lik2 <- fitCopula(joeCopula(),data=U,method="mpl")@ loglik
2*(Log.lik1-1)
```

```
## [1] 52.17909
```

```
2*(Log.lik2-1)
```

```
## [1] 54.43753
```

Non-parametric methods

- Empirical copula
- Empirical Beta copula
- Empirical checkboard copula
- Extreme value copula

Non-parametric methods: Empirical copula

The empirical copula estimator of copula C is

$$C_n(u_1, ..., u_n) = \frac{1}{n}\sum_{i=1}^{n} I(U_{i,n} \leq u) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} I(U_{ij,n} \leq u_j),$$

```
library(copula)
mydist <- mvdc(gumbelCopula(param = 2, dim = 2),
          margins = c("gamma","gamma"),
          paramMargins = list(list(shape = 2,
          rate = 3), list(shape = 4, rate = 5)))
fake.data <- rMvdc(1000, mydist)
U <-pobs(fake.data)
C.n(c(0.6,0.4),X=U)
```

```
## [1] 0.349
```

```
C.n(c(0.8,0.7),X=U)
```

```
## [1] 0.654
```

Non-parametric methods: Empirical Beta copula

The empirical Beta copula estimator of copula C is

$$C_n(u_1, ..., u_n) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d} F_{n,R_{ij}}(u_j), \quad u \in [0,1]^d.$$

```r
library(copula)
U <-pobs(fake.data)
C.n(c(0.6,0.4), X=U, smoothing="beta")
```

```
## [1] 0.3506331
```

```r
C.n(c(0.8,0.7), X=U, smoothing="beta")
```

```
## [1] 0.6539474
```

Non-parametric methods: Empirical checkboard copula

The empirical Beta copula estimator of copula C is

$$C_n(u_1, ..., u_n) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}\min\Big\{\max\{nu_j - R_{ij} + 1, 0\}, 1\Big\}, \quad u \in [0,1]^d.$$

```
library(copula)
cop <- gumbelCopula(param = 2, dim = 2)
fake.data <- rCopula(1000, cop)
U <-pobs(fake.data)
C.n(c(0.6,0.4), X=U, smoothing="checkerboard")
```

```
## [1] 0.352
```

```
C.n(c(0.8,0.7), X=U, smoothing="checkerboard")
```

```
## [1] 0.648
```

Any questions?

This session is finished!