Statistical and computational aspects of nested Archimedean copulas and beyond

Marius Hofert

2013-10-18

Technische Universität München

1 Copulas

Definition: A copula C is a distribution function with $\mathrm{U}[0,1]$ margins.

Theorem 1.1 (Sklar (1959))
$H(x_1,\dots,x_d) = C(F_1(x_1),\dots,F_d(x_d)),\ \ (x_1,\dots,x_d) \in \mathbb{R}^d$

"$\Rightarrow$" Decomposition (estimation, goodness-of-fit)

- $+$ Investigating dependencies $(F_1(X_1),\dots,F_d(X_d)) \sim C$
- $+$ Numerics: reduce dimension of parameter space

"$\Leftarrow$" Composition (finance, sampling, stress testing)

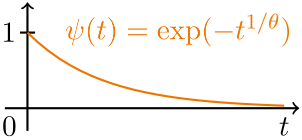
- $+$ Framework for constructing distributions
- $+$ Financial and Insurance Mathematics: Realistic, flexible models

Example: $C(\boldsymbol{u}) = t_{\nu,P}(t_\nu^{-1}(u_1),\dots,t_\nu^{-1}(u_d))$ (t copula; numerics!)

2 Archimedean copulas

Archimedean copula with generator ψ:

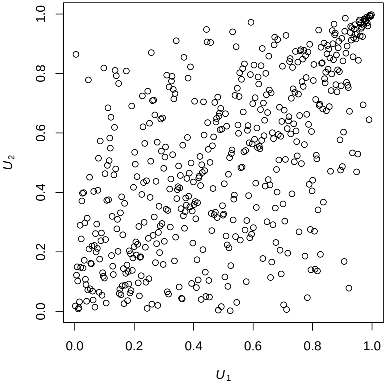
$$C(\boldsymbol{u}) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d))$$



2.1 Properties

+ C is explicit (if ψ^{-1} is)
+ Properties of C can be expressed in terms of ψ
+ $\lambda_L \neq \lambda_U$ possible
+ Examples: AMH, Clayton, Frank, Gumbel, Joe, opC,...

− symmetry



Question: When is $C(\boldsymbol{u}) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d))$ a copula?

Answer: C is a copula in any dimension $d \;\Leftrightarrow\; \psi$ is completely monotone.

Theorem 2.1 (Bernstein (1928))
ψ c.m., $\psi(0) = 1 \;\Leftrightarrow\; \psi(t) = \mathcal{LS}[F](t) = \int_0^\infty \exp(-tv)\, dF(v)$

$\Rightarrow$ Mixture representation:

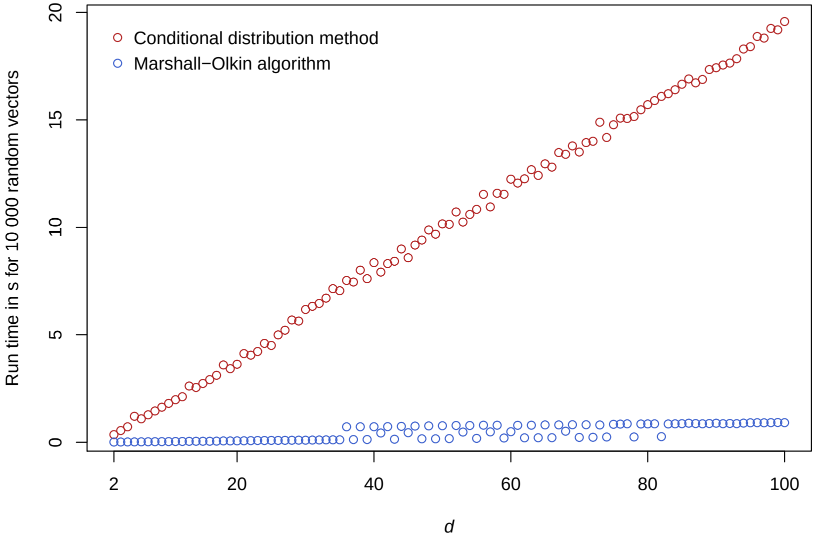
$$C(\boldsymbol{u}) = \psi\Big(\sum_{j=1}^{d} \psi^{-1}(u_j)\Big) = \int_0^\infty \prod_{j=1}^{d} \exp(-v\psi^{-1}(u_j))\, dF(v)$$

$\Rightarrow$ Stochastic representation + Marshall-Olkin:

$$V \sim F = \mathcal{LS}^{-1}[\psi] \;\Rightarrow\; \boldsymbol{U} = \Big(\psi\Big(\frac{E_1}{V}\Big), \dots, \psi\Big(\frac{E_d}{V}\Big)\Big)^{\top} \sim C$$

Fast to sample, independently of d.

Comparison in the "best-case scenario" (Clayton):



2.2 Density evaluation

Statistical challenge: Compute the log-density of a Gumbel copula

General formula for c (trivial):

$$C(\boldsymbol{u}) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d))$$

$$\Rightarrow \; c(\boldsymbol{u}) = (-1)^d \psi^{(d)}\Bigg(\sum_{j=1}^{d} \psi^{-1}(u_j)\Bigg) \cdot \prod_{j=1}^{d} -(\psi^{-1})'(u_j).$$

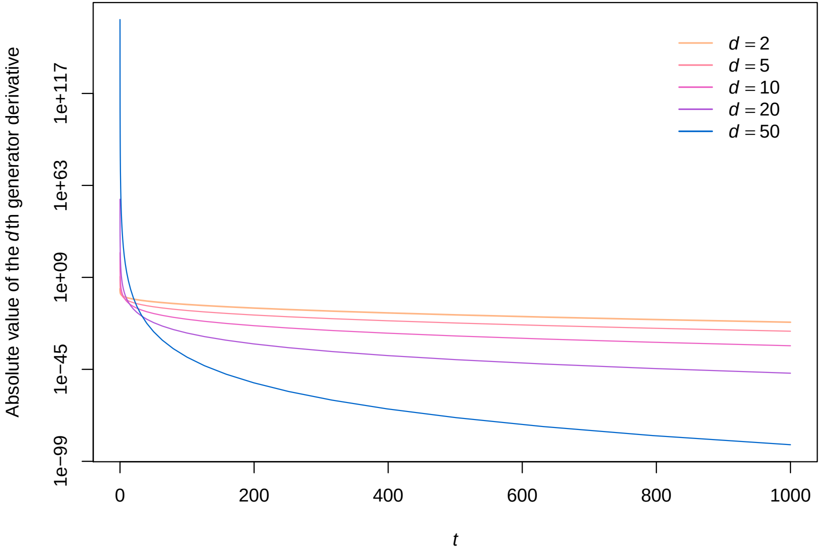
Mathematical challenge: Find $(-1)^d\psi^{(d)}$, $\psi(t) = \exp(-t^\alpha)$, $\alpha = 1/\theta$.

Hofert et al. (2012):

$$(-1)^d \psi^{(d)}(t) = \frac{\psi(t)}{t^d} P(t^\alpha), \quad P(x) = \sum_{k=1}^{d} a_{dk}^{\mathsf{G}}(\alpha) x^k,$$

$$a_{dk}^{\mathsf{G}}(\alpha) = (-1)^{d-k} \sum_{j=k}^{d} \alpha^j s(d,j) S(j,k) = \frac{d!}{k!} \sum_{j=1}^{k} \binom{k}{j} \binom{\alpha j}{d} (-1)^{d-j}$$

Numerical challenge: Computing $(-1)^d\psi^{(d)}(t)$, then $\log(\cdot)$ fails!



One Idea: exp-log-trick

$$\begin{aligned}\log P(x) &= \log \sum_{k=1}^{d} \exp(b_k),\ \ b_k = \log(a_{dk}^{\mathsf{G}}(\alpha)x^k) \\ &= \log\Big(\exp(b_{\mathsf{max}}) \cdot \sum_{k=1}^{d} \exp(b_k - b_{\mathsf{max}})\Big) \\ &= b_{\mathsf{max}} + \log \sum_{k=1}^{d} \exp(b_k - b_{\mathsf{max}})\end{aligned}$$

Advantages: 1) summands are in $(0, 1]$

2) we can work in log-scale

3) can apply a similar trick to $\log a_{dk}^{\mathsf{G}}(\alpha)$

. . . the actual implementation is more difficult, though.

The exp-log-trick is not ideal for all t, d, α. Other methods are:

1) exp-log-trick + $a_{dk}^{\mathsf{G}}(\alpha)$ via

- `direct` (direct);
- `horner` (Horner's scheme);
- $a_{dk}^{\mathsf{G}}(\alpha) = \dfrac{p_{01}^{\mathsf{J}}(d;k)d!}{k!}$ and other methods (`Rmpfr`).

2) $P(x) = (-1)^{d-1} x \sum_{j=1}^{d} \Big(s(d,j) \sum_{k=0}^{j-1} S(j,k+1)(-x)^k \Big) \alpha^j$

- `stirling` (direct; inner sum via Horner's scheme);
- `stirling.horner` (Horner's scheme twice).

3) $P(x) = \sum_{j=1}^{d} s_j \exp\big(\log|(\alpha j)_d| + j \log x + x - \log(j!) + \log F^{\mathsf{Poi}(x)}(d-j)\big)$

- `pois.direct` (direct);
- `pois` (exp-log-trick).

$\Rightarrow$ A good default chooses between these based on t, d, α and run time.

Software challenge: How to check results for correctness? CASs fail!

Example: $\psi^{(50)}(15) = ?$ ($\theta = 5/4$; correct answer: 1056.94)

- Maple 14: 10 628, -29 800,... (chaotic; sign wrong; slow)
- Mathematica 8: – (aborted after 10 min)
- MATLAB 7.11.0: ✓ ($d = 100$: aborted after several min)
- Sage 4.7.1: – (aborted after 10 min) $\Rightarrow$ rarely numerically stable

Note: This is only one evaluation! It has to be done...

1) $n (= 100)$ times for computing the log-likelihood once
2) $m (= 10)$ times for computing MLEs
3) $B (= 1000)$ times within a bootstrap
4) $N (= 1000)$ times to (numerically) show bootstrap convergence
5) for various n, d, θ...

$\Rightarrow$ Parallel computing (more on that later)

Run time aside, what about Monte Carlo?

$$\log((-1)^d\psi^{(d)}(t)) \approx \log\left(\frac{1}{m}\sum_{k=1}^m V_k^d \exp(-V_k t)\right) = \log\left(\frac{1}{m}\sum_{k=1}^m \exp(b_k)\right)$$

$\Rightarrow$ Apply exp-log-trick with $b_k = d\log(V_k) - V_k t$.

Question: Besides increasing run time, does this work? Yes and No!

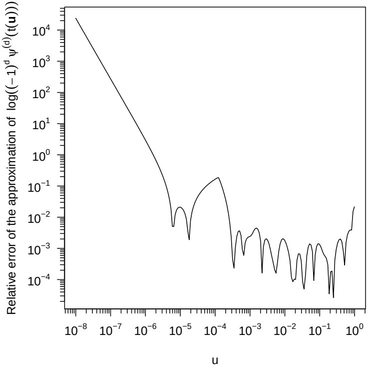
Clayton: Assume $\tau = 0.5$ and $\boldsymbol{u} = (u,\dots,u)^\top$. If u is small, then $t = \sum_{j=1}^d \psi^{-1}(u_j) = d\psi^{-1}(u)$ is large.

Problem: b_k is so small that

$$e^{b_k} = e^{b_k - b_{\max}} = 0.$$

Note: `exp(-746)==0` is `TRUE`!

Solution: Pseudo-observations!



Extensions to more general/flexible models:

1) Outer power Archimedean copulas: $\tilde{\psi}(t) = \psi(t^{1/\beta})$, $\beta \geq 1$

$$(-1)^d \tilde{\psi}^{(d)}(t) = \frac{P(t^{1/\beta})}{t^d}, \quad P(x) = \sum_{k=1}^{d} a_{dk}^{\mathsf{G}}(1/\beta)\psi^{(k)}(x)(-x)^k$$

2) Khoudraji-transformed Archimedean copulas (special case of Liebscher):

$$C(\boldsymbol{u}) = C_\psi(u_1^{\alpha_1}, \dots, u_d^{\alpha_d})\,\Pi(u_1^{1-\alpha_1}, \dots, u_d^{1-\alpha_d})$$

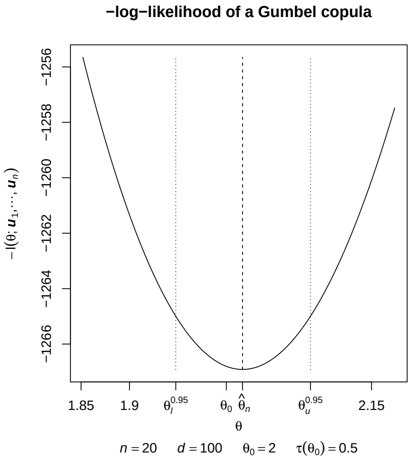
$$c(\boldsymbol{u}) = \sum_{J \subseteq \{1,\dots,d\}} \psi^{(|J|)}\Bigg(\sum_{j=1}^{d} \psi^{-1}(u_j^{\alpha_j})\Bigg) \prod_{j \in J} \alpha_j (\psi^{-1})'(u_j^{\alpha_j}) \prod_{j \notin J} (1-\alpha_j) u_j^{-\alpha_j}$$

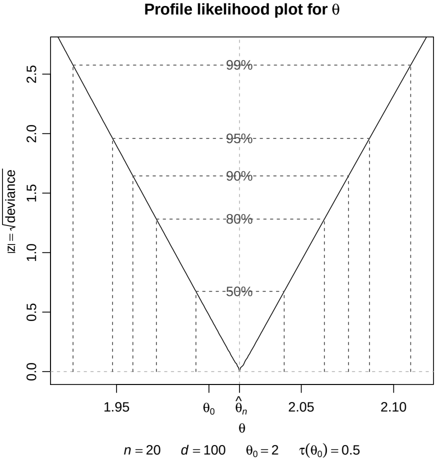
3) Archimedean Sibuya copulas: $C(\boldsymbol{u}) = \dfrac{\psi_V\left(\sum_{j=1}^d f(S_j^-(u_j))\right)}{\prod_{j=1}^d \psi_V\left(f(S_j^-(u_j))\right)}\Pi(\boldsymbol{u})$

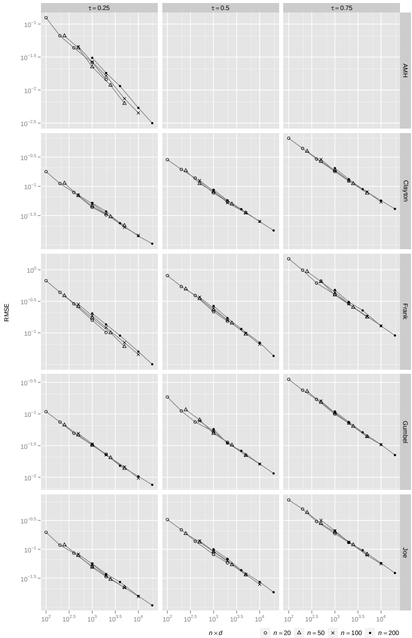
4) Kendall distribution function: $K(t) = \displaystyle\sum_{k=0}^{d-1} \frac{\psi^{(k)}(\psi^{-1}(t))}{k!}(-\psi^{-1}(t))^k$

5) Nested Archimedean copulas: $C(\boldsymbol{u}) = C_0(C_1(\boldsymbol{u}_1), \dots, C_S(\boldsymbol{u}_S))$

If you pay attention to the numerics...







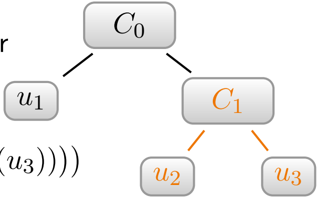
- Striking (numerical) result:

$$\boxed{\mathrm{MSE} \propto \frac{1}{nd}}$$

(known margins)

- See Hofert et al. (2013)

3 Nested Archimedean copulas



Idea: Plug Archimedean copulas into each other

$$\begin{aligned} C(\boldsymbol{u}) &= C_0(u_1, C_1(u_2, u_3)) \\ &= \psi_0(\psi_0^{-1}(u_1) + \psi_0^{-1}(\psi_1(\psi_1^{-1}(u_2) + \psi_1^{-1}(u_3)))) \end{aligned}$$

$\Rightarrow$ Partial asymmetries

Question: When is it a copula?

> **Theorem 3.1 (Joe (1997), McNeil (2008))**
> $\mathring{\psi}_{01}' = (\psi_0^{-1} \circ \psi_1)'$ completely monotone $\Rightarrow C$ is a copula

$\Rightarrow$ New generator: $\psi_{01}(t; v) = \exp(-v\,\mathring{\psi}_{01}(t)) = \mathcal{LS}[F_{01}(\cdot\,; v)](t)$.

Idea: $\psi_{01}(t;v) = \exp(-v\psi_0^{-1}(\psi_1(t)))$, $G(u;v) = \exp(-v\psi_0^{-1}(u))$

Then

$$\begin{aligned} C(\boldsymbol{u}) &= \psi_0(\psi_0^{-1}(u_1) + \psi_0^{-1}(\psi_1(\psi_1^{-1}(u_2) + \psi_1^{-1}(u_3)))) \\ &= \int_0^\infty \exp(-v\psi_0^{-1}(u_1))\exp(-v\psi_0^{-1}(\psi_1(\psi_1^{-1}(u_2) + \psi_1^{-1}(u_3))))\,dF_0(v) \\ &= \int_0^\infty G(u_1;v)\,\psi_{01}(\psi_1^{-1}(u_2) + \psi_1^{-1}(u_3);v)\,dF_0(v) \\ &= \int_0^\infty G(u_1;v)\,\psi_{01}(\psi_{01}^{-1}(G(u_2;v);v) + \psi_{01}^{-1}(G(u_3;v);v);v)\,dF_0(v) \\ &= \int_0^\infty G(u_1;v)\,C_{01}(G(u_2;v), G(u_3;v);v)\,dF_0(v) \end{aligned}$$

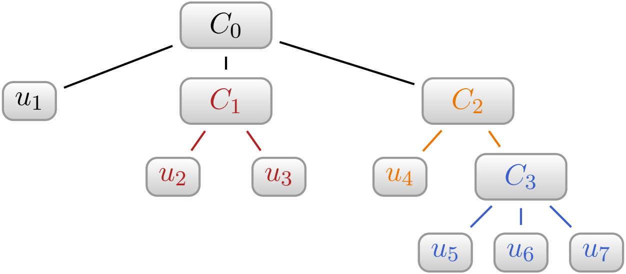
$\Rightarrow$ If C_{01} is a copula, so is C

Examples: AMH ∘ C (c.m.), but also F ∘ C ($d = 3$)

AB (2009): "… $\mathring{\psi}_{01}$ must have completely monotone derivatives…"

3.1 Stochastic representation and sampling

$C(\boldsymbol{u}) = C_0(u_1, C_1(u_2, u_3), C_2(u_4, C_3(u_5, u_6, u_7)))$:



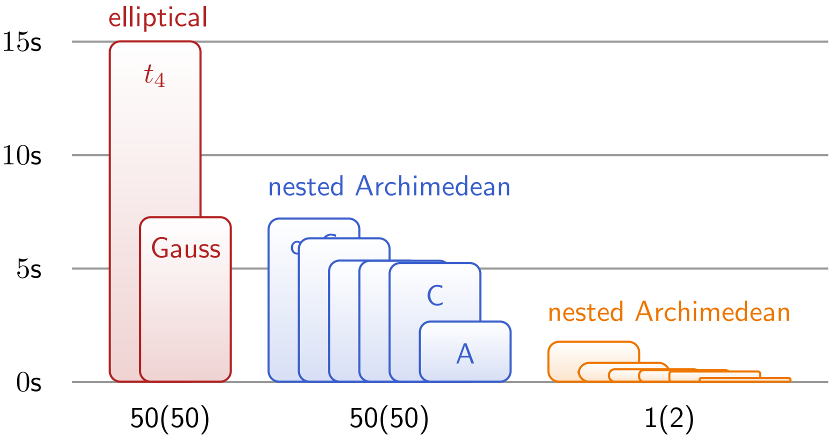
$$\left(\psi_0\Big(\frac{E_1}{V_0}\Big), \psi_1\Big(\frac{E_2}{V_{01}}\Big), \psi_1\Big(\frac{E_3}{V_{01}}\Big), \psi_2\Big(\frac{E_4}{V_{02}}\Big), \psi_3\Big(\frac{E_5}{V_{23}}\Big), \psi_3\Big(\frac{E_6}{V_{23}}\Big), \psi_3\Big(\frac{E_7}{V_{23}}\Big)\right)$$

where $V_0 \sim \mathcal{LS}^{-1}[\psi_0]$ $\quad V_{01}|V_0 \sim \mathcal{LS}^{-1}[\psi_{01}(\cdot\,;V_0)]$

$V_{02}|V_0 \sim \mathcal{LS}^{-1}[\psi_{02}(\cdot\,;V_0)]$ $\quad V_{23}|V_{02} \sim \mathcal{LS}^{-1}[\psi_{23}(\cdot\,;V_{02})]$

$\Rightarrow$ McNeil (2008), Hofert (2012), R package `copula`

Overview: Sampling $100\,000$ $\boldsymbol{U} \sim C = C_0(\cdot\,, C_1(\cdot\,))$



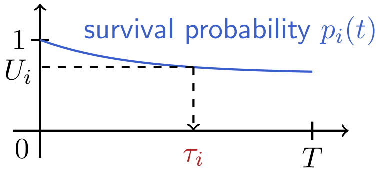
$\Rightarrow$ Fast enough for large-scale simulations (see Hofert and Scherer (2011)). Function evaluations and $\mathrm{Exp}(1)$ are more relevant than F_0 and F_{01}.

3.2 Application to credit risk

Intensity-based default model:

$$p_i(t) = \exp\Big(-\int_0^t \lambda_i(s)\,ds\Big)$$

$$\tau_i = \inf\{t \geq 0 : p_i(t) \leq U_i\}$$



Note: $\lambda_U = 0$ $\Rightarrow$ (Almost) no joint defaults!

Copulas for the triggers $\boldsymbol{U}$:

1) Li (2000): Gaussian (Sibuya (1959): $\lambda_U = 0$)

2) Schönbucher and Schubert (2001): Archimedean ($\lambda_U > 0$)

3) Hofert and Scherer (2011): nested Archimedean ($\lambda_U > 0$, hierarchies)

3.3 Densities of nested Archimedean copulas

In the literature (AB (2009)):

> "… Not even for an EAC it is straightforward to derive the density… For the NAC it is even harder…"

Poor man's approach: $C(\boldsymbol{u}) = C_0(C_1(u_1, u_2), C_2(u_3, u_4))$

$$
\begin{aligned}
c(\boldsymbol{u}) = {} & (\psi_1^{-1})'(u_1)(\psi_1^{-1})'(u_2)\Bigg[\Big\{(\psi_0^{-1})'(C_1)\psi_1'[\psi_1^{-1}(u_1)+\psi_1^{-1}(u_2)]\Big\}^2 \\
& \frac{\partial^2}{\partial u_4 \partial u_3}\psi_0''[\psi_0^{-1}(C_1)+\psi_0^{-1}(C_2)] + \Big((\psi_0^{-1})''(C_1)\{\psi_1'[\psi_1^{-1}(u_1)+\psi_1^{-1}(u_2)]\}^2 \\
& + (\psi_0^{-1})'(C_1)\psi_1''[\psi_1^{-1}(u_1)+\psi_1^{-1}(u_2)]\Big)\frac{\partial^2}{\partial u_4 \partial u_3}\psi_0'[\psi_0^{-1}(C_1)+\psi_0^{-1}(C_2)]\Bigg]
\end{aligned}
$$

$\Rightarrow$ Two more to go… ☹

Statistical challenge: Compute the log-density of NACs

$$C(\boldsymbol{u}) = C_0(C_1(\boldsymbol{u}_1), \dots, C_{d_0}(\boldsymbol{u}_{d_0})), \quad \text{where}$$

$$C_s(\boldsymbol{u}_s) = \psi_s(\psi_s^{-1}(u_{s1}) + \dots + \psi_s^{-1}(u_{sd_s})) = \psi_s(t_s(\boldsymbol{u}_s)).$$

Idea:

$$\begin{aligned} C(\boldsymbol{u}) &= \int_0^\infty \exp\Bigg(-v_0 \sum_{s=1}^{d_0} \mathring{\psi}_{0s}(t_s(\boldsymbol{u}_s))\Bigg)\, dF_0(v_0) \\ &= \int_0^\infty \prod_{s=1}^{d_0} \psi_{0s}(t_s(\boldsymbol{u}_s); v_0)\, dF_0(v_0), \end{aligned}$$

where $\mathring{\psi}_{0s} = \psi_0^{-1} \circ \psi_s$ and $\psi_{0s}(t; v_0) = \exp(-v_0 \mathring{\psi}_{0s}(t))$. Therefore,

$$c(\boldsymbol{u}) = \int_0^\infty \prod_{s=1}^{d_0} \psi_{0s}^{(d_s)}(t_s(\boldsymbol{u}_s); v_0)\, dF_0(v_0) \cdot \prod_{s=1}^{d_0} \prod_{j=1}^{d_s} (\psi_s^{-1})'(u_{sj}).$$

Tasks: Find $\psi_{0s}^{(d_s)}(\cdot\,;\, v_0)$ and compute $\int_0^\infty \dots\, dF_0(v_0)$.

Theorem 3.2 (Hofert and Pham (2013))

1) Let $a_{s,nk}(t) = B_{n,k}(\mathring{\psi}_{0s}'(t), \dots, \mathring{\psi}_{0s}^{(n-k+1)}(t))$. Then

$$\psi_{0s}^{(n)}(t; v_0) = \psi_{0s}(t; v_0) \sum_{k=1}^{n} a_{s,nk}(t)(-v_0)^k.$$

2) Let $b_{\boldsymbol{d},k}^{d_0}(\boldsymbol{t}(\boldsymbol{u})) = \sum_{\boldsymbol{j} \in \mathcal{Q}_{\boldsymbol{d},k}^{d_0}} \prod_{s=1}^{d_0} a_{s,d_s j_s}(t_s(\boldsymbol{u}_s))$, where

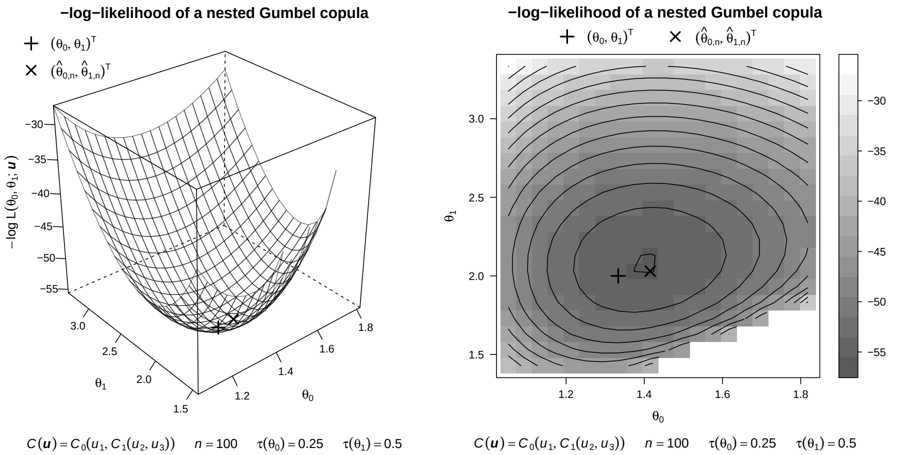
$$\mathcal{Q}_{\boldsymbol{d},k}^{d_0} = \Big\{ \boldsymbol{j} \in \mathbb{N}^{d_0} : \sum_{s=1}^{d_0} j_s = k,\ j_s \le d_s,\ s \in \{1, \dots, d_0\} \Big\}.$$

Then

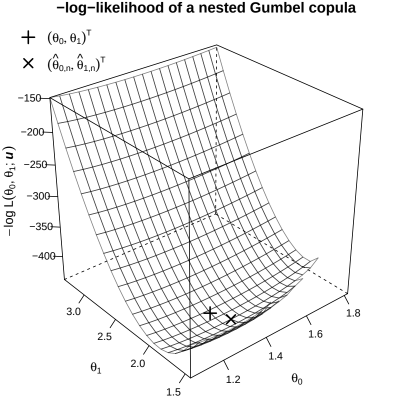
$$c(\boldsymbol{u}) = \Bigg(\sum_{k=d_0}^{d} b_{\boldsymbol{d},k}^{d_0}(\boldsymbol{t}(\boldsymbol{u})) \psi_0^{(k)}(\psi_0^{-1}(C(\boldsymbol{u}))) \Bigg) \cdot \prod_{s=1}^{d_0} \prod_{j=1}^{d_s} (\psi_s^{-1})'(u_{sj}).$$

A picture is worth a thousand words...

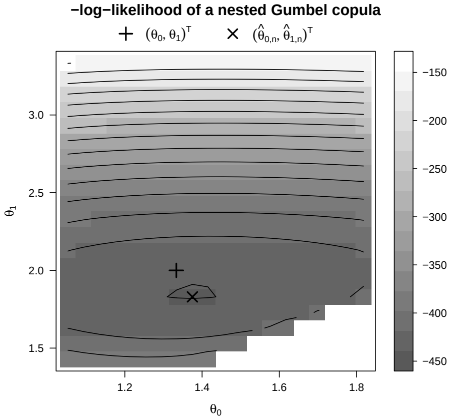
-log-likelihood of a nested Gumbel copula $C(\boldsymbol{u}) = C_0(u_1, C_1(u_2, u_3))$:



-log-likelihood of a nested Gumbel copula $C(\boldsymbol{u}) = C_0(u_1, C_1(u_2, \dots, u_{10}))$:



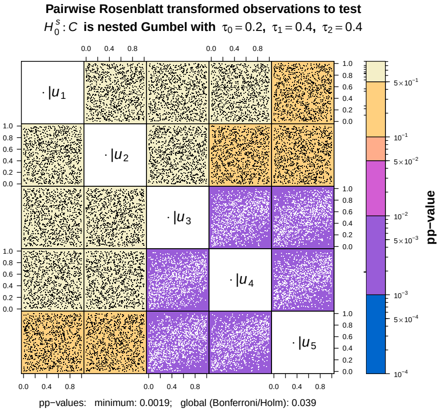
$C(\boldsymbol{u}) = C_0(u_1, C_1(u_2, \dots, u_{10}))$ $n = 100$ $\tau(\theta_0) = 0.25$ $\tau(\theta_1) = 0.5$

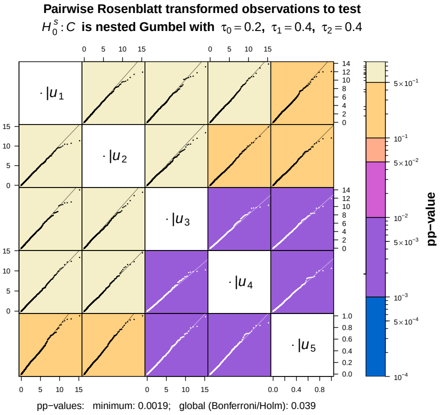


$C(\boldsymbol{u}) = C_0(u_1, C_1(u_2, \dots, u_{10}))$ $n = 100$ $\tau(\theta_0) = 0.25$ $\tau(\theta_1) = 0.5$

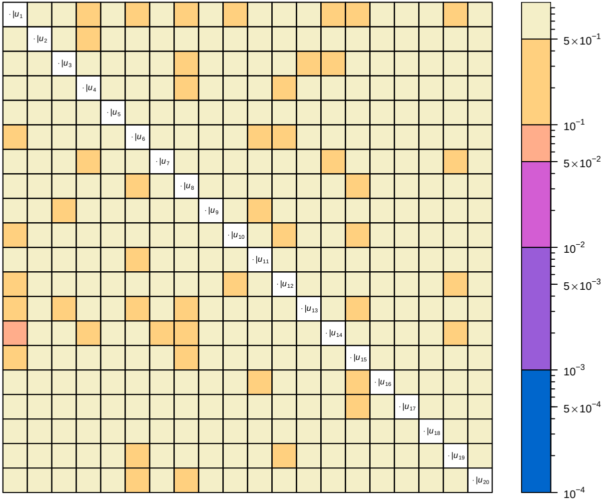
3.4 Graphical goodness-of-fit

$C_0(C_1(u_1, u_2), C_2(u_3, u_4, u_5))$ (Gumbel with $\tau_0 = 0.2$, $\tau_1 = 0.4$, $\tau_2 = 0.6$)





Pairwise Rosenblatt transformed pseudo−observations to test $H_0^c: C$ is t



pp−values: minimum: 0.084; global (Bonferroni/Holm): 1

- SMI log-returns
- 2011-09-09–2012-03-28 (141d)
- MLE for ν (pairw. tau for P)
- `demo(gof_graph)`

4 Large-scale simulation studies

- joint work with Martin Mächler (SfS)
- `simsalapar` (simulations simplified and launched parallel)
- For students (master/PhD), researchers (new models), practitioners (time constraints, validating internal models)

A simulation consists of the following parts:

1) Setup:
   - Scientific problem/question
   - Translating it to R (determining input variables and their "type")
   - Implementing the main, problem specific function (`doOne()`)
2) Conducting the simulation: sequentially? in parallel? nodes or cores?
3) Analyzing the results: Computing and presenting statistics (with tables or graphics)

Example: Computing VaR_α via Monte Carlo

Consider the value of a portfolio of stocks $S_{t,1},\dots,S_{t,d}$

$$V_t = \sum_{j=1}^{d} \beta_j S_{t,j}.$$

In terms of the risk-factor changes

$$X_{t+1,j} = \log(S_{t+1,j}/S_{t,j})$$

the one period ahead loss is

$$L_{t+1} = -(V_{t+1} - V_t) = -\sum_{j=1}^{d} w_{t,j}(\exp(X_{t+1,j}) - 1), \quad w_{t,j} = \beta_j S_{t,j}.$$

Scientific problem: Compute $\mathrm{VaR}_\alpha(L_{t+1})$ via Monte Carlo under various setups and investigate its behavior.

Translating this...

... to R:

Variable	expression	type	value
`n.sim`	N_{sim}	N	32
`n`	n	grid	64, 256
`d`	d	grid	5, 20, 100, 500
`varWgts`	$\mathbf{w}$	frozen	1, 1, 1, 1
`qF`	F^{-1}	frozen	qF
`family`	C	grid	Clayton, Gumbel
`tau`	τ	grid	0.25, 0.50
`alpha`	α	inner	0.950, 0.990, 0.999

```
require("simsalapar")
varList ←
    varlist( # constructor for an object of class 'varlist'
        n.sim = list(type="N", expr = quote(N[sim]), value = 32), # replications
        n = list(type="grid", value = c(64, 256)), # sample size
        d = list(type="grid", value = c(5, 20, 100, 500)), # dimension and weights
        varWgts = list(type="frozen", expr = quote(bold(w)),
                       value = list("5"=1, "20"=1, "100"=1, "500"=1)),
        qF = list(type="frozen", expr = quote(F^{-1}), value=list(qF=qnorm)), # margins
        family=list(type="grid", expr = quote(C), value = c("Clayton", "Gumbel")), # C
        tau = list(type="grid", value = c(0.25, 0.5)), # Kendall's tau
        alpha = list(type="inner", value = c(0.95, 0.99, 0.999))) # confidence levels
toLatex(varList) # method for constructing the above table
```

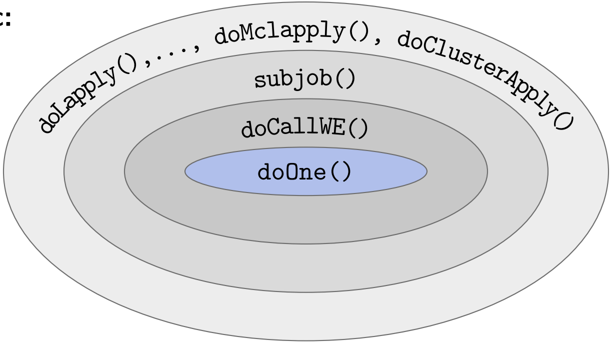
The workhorse: (Computation for **one** set of variables)

```
doOne <- function(n, d, qF, family, tau, alpha, varWgts, names=FALSE)
{
    ## checks (and load required packages here for parallel computing later on)
    w <- varWgts[[as.character(d)]]
    stopifnot(require(copula), # load 'copula'
              sapply(list(w, alpha, tau, d), is.numeric)) # sanity checks

    ## simulate risk-factor changes (if defined outside doOne(), use
    ## doOne <- local({...}) construction as in some of simsalapar's demos)
    simRFC <- function(n, d, qF, family, tau) {
        ## define the copula of the risk factor changes
        theta <- getAcop(family)@iTau(tau) # determine copula parameter
        cop <- onacopulaL(family, list(theta, 1:d)) # define the copula
        ## sample the meta-copula-model for the risk-factor changes X
        qF(rCopula(n, cop)) # simulate via Sklar's Theorem
    }
    X <- simRFC(n, d=d, qF=qF[["qF"]], family=family, tau=tau) # simulate X

    ## compute the losses and estimate VaR_alpha(L)
    L <- -rowSums(expm1(X) * matrix(rep(w, length.out=d),
                                    nrow=n, ncol=d, byrow=TRUE)) # losses
    quantile(L, probs=alpha, names=names) # empirical quantile as VaR estimate
}
```

The magic:



`do*()`: calls `subjob()` sequentially or in parallel (nodes or cores!)

`subjob()`: computes a sub-job (line in virtual grid), `.Random.seed`

`doCallWE()`: catching warnings, errors, run time

`doOne()`: computing a value (for one set of variables; numeric array)

```
res ← doClusterApply(varList, sfile="res.rds", doOne=doOne, names=TRUE) # same interface
```

Extract results (value, errors, warnings, run times):

```
> val  ← getArray(res) # array of values
> err  ← getArray(res, "error") # array of error indicators
> warn ← getArray(res, "warning") # array of warning indicators
> time ← getArray(res, "time") # array of user times in ms
```

A quick look at the errors:

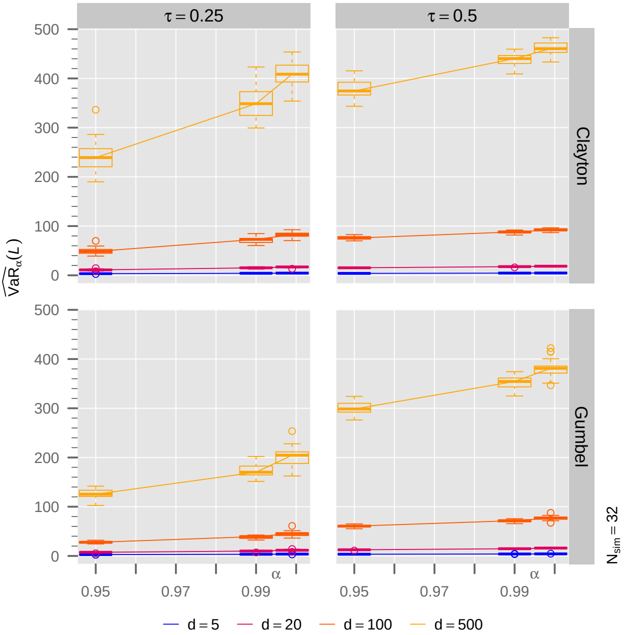
```
> ftable(100 * err, row.vars=c("family", "d"), col.vars=c("tau", "n")) # % of errors
```

```
            tau 0.25     0.50
            n      64 256   64 256
family  d
Clayton 5           0   0    0   0
        20          0   0    0   0
        100         0   0    0   0
        500         0   0    0   0
Gumbel  5           0   0    0   0
        20          0   0    0   0
        100         0   0    0   0
        500         0   0    0   0
```

(Much) more sophisticated tools...

```r
ft <- ftable(fres, row.vars = c("family", "n", "d"), col.vars = c("tau", "alpha"))
tabL <- toLatex(ft, vList = varList, fontsize = "scriptsize", caption = "...")
```

		τ	0.25			0.50		
C	n	$d \mid \alpha$	95%	99%	99.9%	95%	99%	99.9%
Clayton	64	5	3.1 (0.4)	3.8 (0.4)	4.0 (0.5)	3.6 (0.3)	4.2 (0.2)	4.4 (0.2)
		20	10.6 (1.4)	13.5 (1.5)	14.8 (2.2)	14.2 (1.6)	16.7 (1.0)	17.4 (1.0)
		100	46.1 (9.1)	63.5 (11.6)	68.5 (13.6)	70.7 (8.6)	83.7 (3.9)	86.7 (4.2)
		500	224.8 (50.6)	307.8 (61.5)	336.0 (66.8)	350.0 (40.5)	418.6 (22.3)	434.0 (21.4)
	256	5	3.2 (0.2)	4.1 (0.2)	4.4 (0.2)	3.9 (0.2)	4.4 (0.1)	4.6 (0.1)
		20	10.9 (1.0)	15.3 (1.2)	17.0 (0.9)	15.3 (0.7)	17.6 (0.5)	18.5 (0.6)
		100	49.0 (5.5)	72.1 (7.7)	82.5 (4.8)	76.0 (3.4)	87.9 (2.7)	92.3 (3.0)
		500	240.4 (27.0)	349.7 (35.3)	408.5 (24.3)	378.8 (17.4)	439.4 (12.7)	461.7 (14.2)
Gumbel	64	5	2.7 (0.3)	3.3 (0.4)	3.4 (0.5)	3.3 (0.3)	3.8 (0.3)	4.0 (0.2)
		20	7.3 (1.1)	9.4 (1.2)	10.1 (1.5)	12.2 (0.6)	14.0 (1.2)	14.6 (1.2)
		100	26.0 (4.2)	35.8 (4.7)	38.5 (5.6)	57.7 (5.1)	67.7 (4.8)	70.3 (5.4)
		500	117.2 (12.5)	154.4 (19.0)	167.5 (18.2)	288.2 (18.0)	333.7 (23.0)	347.9 (20.7)
	256	5	2.7 (0.2)	3.3 (0.2)	3.7 (0.2)	3.4 (0.2)	3.9 (0.1)	4.2 (0.1)
		20	7.4 (0.5)	9.9 (0.8)	11.5 (0.9)	12.5 (0.4)	14.7 (0.7)	16.0 (0.6)
		100	27.8 (2.8)	38.4 (3.1)	44.7 (3.2)	60.4 (2.3)	70.9 (2.5)	76.9 (3.5)
		500	126.8 (10.3)	171.9 (11.2)	202.3 (13.5)	299.1 (13.7)	353.8 (13.2)	380.0 (9.7)



- `mayplot()`
- more visible than with tables
- new theoretical insights
- a lot more "behind the scenes"

Outlook: The Λ_U inversion problem

Federico Degen (Head Risk Modeling and Quantification, Zurich):

Question: How to construct a four-dimensional distribution with matrix of pairwise upper tail dependence coefficients

$$\Lambda_U = \begin{pmatrix} 1 & 0 & 0 & \alpha \\ 0 & 1 & 0 & \alpha \\ 0 & 0 & 1 & \alpha \\ \alpha & \alpha & \alpha & 1 \end{pmatrix}, \quad \alpha \in (0,1) \ ?$$

Joe (1997) $\Rightarrow$ There is no such model for $\alpha > 1/2$.

Open problems:

- ... Is this bound sharp? How does it extend to general d?
- ... How does it change for different Λ_U?
- ... How can one construct a corresponding d-dimensional copula?

Interesting connection with hierarchical copulas:

$$\Lambda_U = \begin{pmatrix} 1 & 0 & 0 & \alpha \\ 0 & 1 & 0 & \alpha \\ 0 & 0 & 1 & \alpha \\ \alpha & \alpha & \alpha & 1 \end{pmatrix}, \quad \alpha \in (0,1).$$

Idea: $C(\boldsymbol{u}) = C_0(C_1(u_1, u_2, u_3), u_4)$ with $\lambda_U = \alpha$ and $\lambda_U = 0 \;\Rightarrow\; \Lambda_U$!

Note:

- No model with $\lambda_U < \lambda_U$ is known.
- Deeper problem: When is $C(\boldsymbol{u}) = C_0(C_1(\boldsymbol{u}_1), \dots, C_{d_0}(\boldsymbol{u}_{d_0}))$ a valid copula?
- Numerical tests indicate: Not necessarily, but it can...
- What about densities? Connection with Hofert and Pham (2013)?

Other research projects

QRM:

- Superadditivity of VaR in portfolios of bonds (Alexander J. McNeil)

EVT:

- OpRisk model depending on time and covariates; data from Willis Re (Valérie Chavez-Demoulin, Paul Embrechts) $\Rightarrow$ `QRM, demo("game")`
- Densities of EVC (Gabriel Doyon)

Dependence modeling:

- GoF ($d \gg 2$) via radial parts (Johanna Nešlehová, Christian Genest)
- Tilted Archimedean copulas (Johanna Ziegel)
- Multi-frailty copulas (Alexander J. McNeil)

R packages: `copula`, `simsalapar`; contributing to `QRM`, `mvtnorm`

Thank you for your attention

Technische Universität München