Improved Algorithms for Computing Worst VaR: Numerical Challenges and the ARA

Marius Hofert

(joint work with A. Memartoluie, D. Saunders, T. Wirjanto)

2015-10-02

Department of Statistics and Actuarial Science

UNIVERSITY OF WATERLOO

Disclaimer:

- This is my first work around computing worst(/best) Value-at-Risk.
- I am not an expert on the theory for computing these bounds.
- I will address practical aspects (R: Pkg `qrmtools`, `demo(VaR_bounds)`)

Recall: $H(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$ (Sklar's Theorem)

The problem: Computing worst(/best) VaR

We are given (one-period ahead) losses $L_1 \sim F_1, \dots, L_d \sim F_d$ (e.g., based on fitted $F_1, \dots, F_d$) with known margins and unknown copula C. Consider

$$L^+ = \sum_{j=1}^{d} L_j \text{ and } \mathrm{VaR}_\alpha(L^+) = F_{L^+}^{-}(\alpha) = \inf\{x \in \mathbb{R} : F_{L^+}(x) \geq \alpha\}.$$

Question: How to compute bounds $\underline{\mathrm{VaR}}_\alpha(L^+)$, $\overline{\mathrm{VaR}}_\alpha(L^+)$ on $\mathrm{VaR}_\alpha(L^+)$?
(i.e., the best and worst $\mathrm{VaR}_\alpha(L^+)$ over the set of all copulas)
We will focus on $\overline{\mathrm{VaR}}_\alpha(L^+)$.

We focus on two cases:

1) The homogeneous case (i.e., $F_1 = \dots = F_d =: F$):
   - The dual bound approach (see Puccetti and Rüschendorf (2013), Embrechts et al. (2013, Prop. 4))
   - Wang's approach (see Embrechts et al. (2014, Prop. 1))
2) The inhomogeneous case: The Rearrangement Algorithm (RA; see Puccetti and Rüschendorf (2012), Embrechts et al. (2013))

Not discussed here are, e.g.:

- Bernard et al. (2013) and Bernard et al. (2014) (partial information known about C)
- Bernard and McLeish (2015), Jakobsons et al. (2015) (alternatives to the RA)
- Other references (quickly growing in this field).

1 Solutions in the homogeneous case

Wang's approach for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

- Assume that $F = F_1$ has a decreasing density on $[\beta, \infty)$.
- Let $a_c = \alpha + (d-1)c$, $b_c = 1 - c$ and

$$\bar{I}(c) := \frac{1}{b_c - a_c} \int_{a_c}^{b_c} F^-(y)\,dy, \quad c \in (0, (1-\alpha)/d]$$

Embrechts et al. (2014, Prop. 1) and Wang et al. (2013, Cor. 3.7):

» For $L \sim F$ and $\alpha \in [F(\beta), 1)$,

$$\overline{\mathrm{VaR}}_\alpha(L^+) = d\,\mathbb{E}[L \mid L \in [F^-(a_c), F^-(b_c)]] \underset{\text{Subs.}}{=} d\bar{I}(c),$$

where c is the smallest number in $(0, (1-\alpha)/d]$ such that

$$\bar{I}(c) \geq \frac{d-1}{d} F^-(a_c) + \frac{1}{d} F^-(b_c).$$

Algorithm (Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ based on Wang's approach; `worst_VaR_hom(..., method="Wang")`)

1) Specify an initial interval $[c_l, c_u]$ with $0 \le c_l < c_u < (1-\alpha)/d$.
2) Root-finding in c: Iterate over $c \in [c_l, c_u]$ until a c^* is found for which
$$h(c^*) := \bar{I}(c^*) - \Big(\frac{d-1}{d}F^-(a_{c^*}) + \frac{1}{d}F^-(b_{c^*})\Big) = 0.$$
3) Then return $(d-1)F^-(a_{c^*}) + F^-(b_{c^*})$.

- We only need to know the quantile function F^- to compute $\overline{\mathrm{VaR}}_\alpha(L^+)$.
- The numerical integration (for $\bar{I}$) is typically straightforward; explicit for $\mathrm{Par}(\theta)$ margins.
- It remains unclear how to choose $[c_l, c_u]$ (open problem in general):
  - c_l: $h(0) = -\infty$ (fine) but also undefined ($\infty - \infty$; for $\mathrm{Par}(\theta \in (0,1])$)
  - c_u: Numerically problematic: $h((1-\alpha)/d) \underset{\text{l'H.}}{=} 0$

How can we choose c_l and c_u for $F = \mathrm{Par}(\theta)$?

Proposition (c_l, c_u, `worst_VaR_hom(..., method="Wang.Par")`)

The initial interval end points c_l and c_u can be chosen as

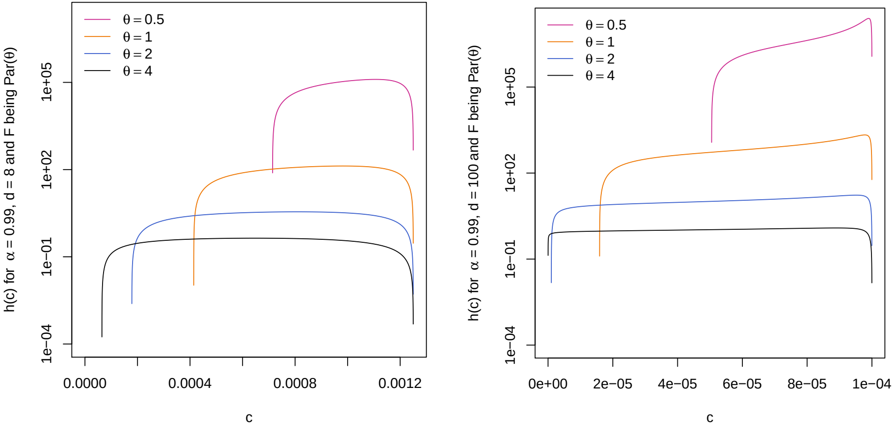
$$c_l = \begin{cases} \frac{(1-\theta)(1-\alpha)}{d}, & \text{if } \theta \in (0,1), \\ \frac{1-\alpha}{(d+1)^{\frac{e}{e-1}}+d-1}, & \text{if } \theta = 1, \\ \frac{1-\alpha}{(d/(\theta-1)+1)^{\theta}+d-1}, & \text{if } \theta \in (1,\infty), \end{cases} \qquad c_u = \begin{cases} \frac{(1-\alpha)(d-1+\theta)}{(d-1)(2\theta+d)}, & \text{if } \theta \neq 1, \\ \frac{1-\alpha}{3d/2-1}, & \text{if } \theta = 1. \end{cases}$$

Proof (idea).

- c_l: Rewrite $h(c) = 0 \Leftrightarrow h_2(x_c) = 0$ for $x_c = (1-\alpha)/c - (d-1)$ and $h_2(x) = (\frac{d}{1-\theta} - 1)x^{-\frac{1}{\theta}+1} - (d-1)x^{-\frac{1}{\theta}} + x - (d\frac{\theta}{1-\theta} + 1)$, $x \in [1,\infty)$. Separately for $\theta \in (0,1)$, $\theta = 1$ and $\theta \in (1,\infty)$, approximate h_2 from below by an invertible function with a root $x_c > 1$; then solve for c.
- c_u: The inflection point of h_2 is a lower bound x_c on the root of h_2; then solve for c. □

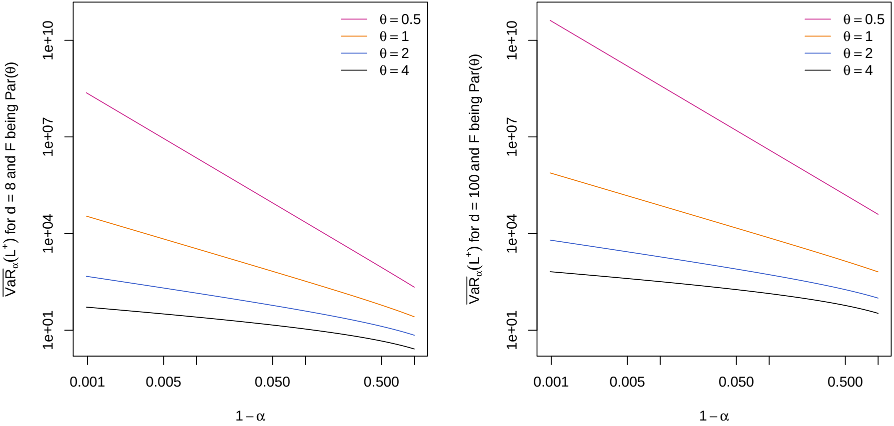
Example ($\overline{\mathrm{VaR}}_\alpha(L^+)$ for $\mathrm{Par}(\theta)$ risks)

Consider $F = \mathrm{Par}(\theta)$ and $\alpha = 0.99$ and plot the objective function $h(c)$ for $d = 8$ (left) and $d = 100$ (right):



(Values $h(c) \leq 0$ have been omitted due to log-scale)

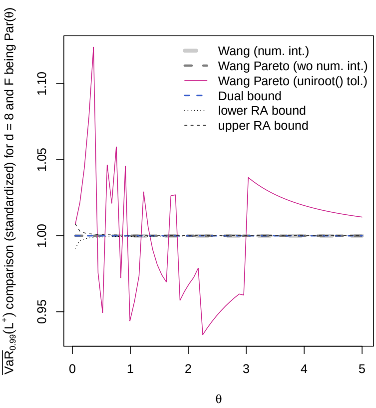
$\overline{\mathrm{VaR}}_\alpha(L^+)$ for various α, θ and $d = 8$ (left) and $d = 100$ (right):

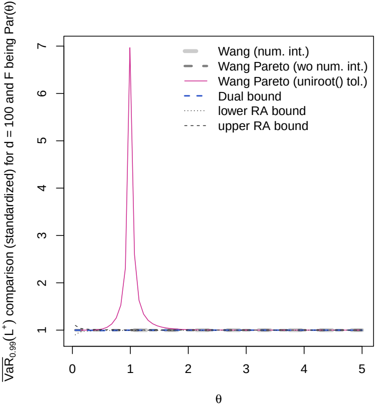


- Nice, right?
- Anything else?

Example (Comparison for $\mathrm{Par}(\theta)$ risks)

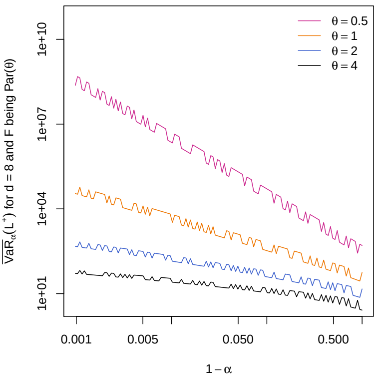
- Wang's approach: with/without num. integration for $\bar{I}$; without num. integration and `uniroot()`'s default tolerance
- Dual bound approach: Numerically trickier... two nested root-findings
- Lower/upper bound RA bounds (results standardized by the h_2 approach)

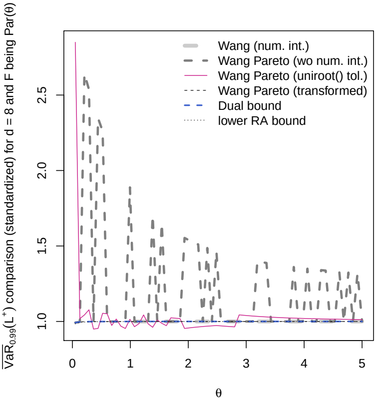




Remark/summary (Word of warning; may apply beyond $\mathrm{Par}(\theta)$)

1) As just seen, the tolerance of `uniroot()` is critical; see below (right)
2) Without c_u: see (left/right) for $h((1-\alpha)/d) =$ `.Machine$double.xmin`





⇒ These are things that are not recognized unless thoroughly tested!

2 The Rearrangement Algorithm

- For the inhomogeneous case for computing ($\underline{\mathrm{VaR}}_\alpha(L^+)$ and) $\overline{\mathrm{VaR}}_\alpha(L^+)$
- The theoretical convergence of $\overline{s}_N - \underline{s}_N \to 0$ is an open problem.
- We focus on practical aspects, not the theory.

2.1 How the RA works

- Two columns $\boldsymbol{a}, \boldsymbol{b}$ are *oppositely ordered* if $(a_i - a_j)(b_i - b_j) \leq 0 \;\forall\, i, j$.
- Row-sum operator $s(X) = \min_{1\leq i\leq N} \sum_{1\leq j\leq d} x_{ij}$

Algorithm (RA for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$)

1) Fix $\alpha \in (0,1)$, $F_1^-, \dots, F_d^-$, $N \in \mathbb{N}$ (# of discr. points), $\varepsilon \geq 0$ (tol.)
2) Compute the lower bound $\underline{s}_N$:
   2.1) Define the (N, d)-matrix $\underline{X}^\alpha = \left(F_j^-\left(\alpha + \frac{(1-\alpha)(i-1)}{N}\right)\right)_{i,j}$.
   2.2) Randomly permute each column of $\underline{X}^\alpha$ (to avoid $\overline{s}_N - \underline{s}_N \nrightarrow 0$)

2.3) Iterate over each column of $\underline{X}^{\alpha}$ and permute it so that it becomes oppositely ordered to the sum of all others $\Rightarrow$ Matrix $\underline{Y}^{\alpha}$

2.4) Repeat Step 2.3) until $s(\underline{Y}^{\alpha}) - s(\underline{X}^{\alpha}) \le \varepsilon$, then set $\underline{s}_N = s(\underline{Y}^{\alpha})$.

3) Compute the upper bound $\overline{s}_N$: Similarly as in Step 2), but based on $\overline{X}^{\alpha} = \Big(F_j^{-}\big(\alpha + \frac{(1-\alpha)i}{N}\big)\Big)_{i,j}$, compute $\overline{s}_N = s(\overline{Y}^{\alpha})$.

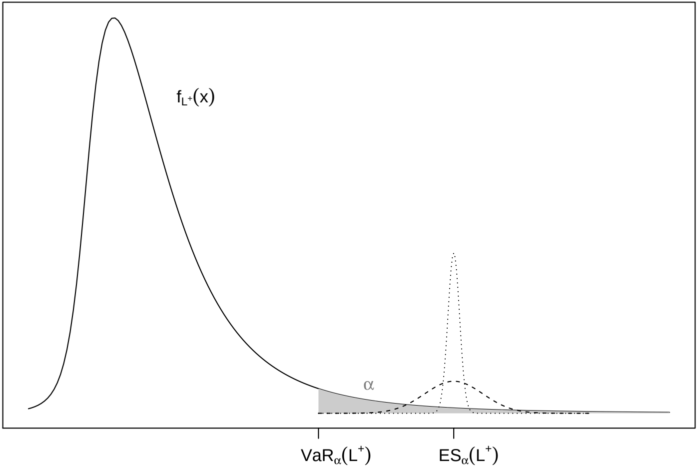
4) Return $(\underline{s}_N,\ \overline{s}_N)$ (*rearrangement range*; taken as $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ bounds)

- **Goal:** Solving the *maximin problem* (minimax for $\underline{\mathrm{VaR}}_{\alpha}$). This can fail, though; see Haus (2014, Lemma 6) for a counter-example.

- **Intuition:** Obtaining a *completely mixable matrix* (row sums constant). This minimizes the variance of $L^+ | L^+ > F_{L^+}^{-}(\alpha)$ to concentrate more of the $1-\alpha$ mass of F_{L^+} in its tail. $\Rightarrow \mathrm{VaR}_{\alpha}(L^+) \uparrow$

A picture is worth a thousand words...

$$\mathrm{VaR}_\alpha(L^+) \le \mathrm{ES}_\alpha(L^+) \underset{L^+ \text{ cont.}}{=} \mathbb{E}[L^+ \mid L^+ > \mathrm{VaR}_\alpha(L^+)]$$



Ideally: $F_1, \dots, F_d$ *jointly mixable* $\Rightarrow \mathbb{P}(L_1 + \cdots + L_d = c) = 1$, $c \in \mathbb{R}$ (in the tail).

Example

1) Where it works (to compute the optimum of the maximin problem):

$$\begin{pmatrix}1&1&1\\2&3&2\\3&5&4\\4&7&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-1}=\begin{pmatrix}2\\5\\9\\15\end{pmatrix}} \begin{pmatrix}4&1&1\\3&3&2\\2&5&4\\1&7&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-2}=\begin{pmatrix}5\\5\\6\\9\end{pmatrix}} \begin{pmatrix}4&7&1\\3&5&2\\2&3&4\\1&1&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-3}=\begin{pmatrix}11\\8\\5\\2\end{pmatrix}}$$

$$\begin{pmatrix}4&7&1\\3&5&2\\2&3&4\\1&1&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-1}=\begin{pmatrix}8\\7\\7\\9\end{pmatrix}} \begin{pmatrix}2&7&1\\4&5&2\\3&3&4\\1&1&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-2}=\begin{pmatrix}3\\6\\7\\9\end{pmatrix}} \begin{pmatrix}2&7&1\\4&5&2\\3&3&4\\1&1&8\end{pmatrix} \overset{\Longrightarrow}{\textstyle\sum_{-3}=\begin{pmatrix}9\\9\\6\\2\end{pmatrix}}$$

$$\begin{pmatrix}2&7&2\\4&5&1\\3&3&4\\1&1&8\end{pmatrix} \checkmark \overset{\Longrightarrow}{\textstyle\sum=\begin{pmatrix}11\\10\\10\\10\end{pmatrix}} \widehat{\mathrm{VaR}}_\alpha(L^+) \approx 10$$

2) Where it fails (to compute the optimum of the maximin problem):

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{pmatrix} \underset{\sum_{-1}=\left(\begin{smallmatrix}2\\4\\6\end{smallmatrix}\right)}{\Longrightarrow} \begin{pmatrix} 3 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 3 & 3 \end{pmatrix} \underset{\sum_{-2}=\left(\begin{smallmatrix}4\\4\\4\end{smallmatrix}\right)}{\Longrightarrow} \begin{pmatrix} 3 & 3 & 1 \\ 2 & 2 & 2 \\ 1 & 1 & 3 \end{pmatrix} \quad \checkmark$$

$$\underset{\sum=\left(\begin{smallmatrix}7\\6\\5\end{smallmatrix}\right)}{\Longrightarrow} \widehat{\mathrm{VaR}}_\alpha(L^+) \approx 5 < 6 \quad \underset{\sum=\left(\begin{smallmatrix}5\\6\\7\end{smallmatrix}\right)}{\Longrightarrow} \text{for} \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

Question (Toronto, 2014; Zurich 2015):
"How to choose $N \in \mathbb{N}$ and $\varepsilon > 0$?"

- No real guidance given in papers. Embrechts et al. (2013, Table 3): Chosen $\varepsilon = 0.1$ is roughly 0.000004% of the computed $\overline{\mathrm{VaR}}_{0.99}(L^+)$.
- Concerning ε, there are two problems:
  1) It would be more natural to use relative tolerances, which guarantee that the change in the minimal row sum from $\underline{X}^\alpha$ ($\overline{X}^\alpha$) to $\underline{Y}^\alpha$ ($\overline{Y}^\alpha$) is of the right order.
  2) ε is only used for checking *individual* "convergence" of $\underline{s}_N$ and of $\overline{s}_N$. There is no guarantee that $\underline{s}_N$ and $\overline{s}_N$ are *jointly* close.
- Also, the algorithm should return more useful information, e.g., 1) $|(\overline{s}_N - \underline{s}_N)/\overline{s}_N|$; 2) the individual tolerances reached for $\underline{s}_N, \overline{s}_N$; 3) the number of iterations used; 4) the row sums after each iteration; or 5) the number of oppositely ordered columns; see `RA()` and `ARA()`.

2.2 Empirical performance under various scenarios

- As studies, we consider the following:

  Study 1: $N \in \{2^7, 2^8, \dots, 2^{17}\}$ and $d = 20$

  Study 2: $N = 256$ and $d \in \{2^2, 2^3, \dots, 2^{10}\}$ (not considered further)

- In each study we investigate the following cases (based on $\alpha = 0.99$, $\varepsilon = 0.001$ and Pareto $F_j(x) = 1 - (1+x)^{-\theta_j}$ margins):

  Case HH : $\theta_1, \dots, \theta_d$ equidistant in $[0.6,\ 0.4]$ (all heavy-tailed)

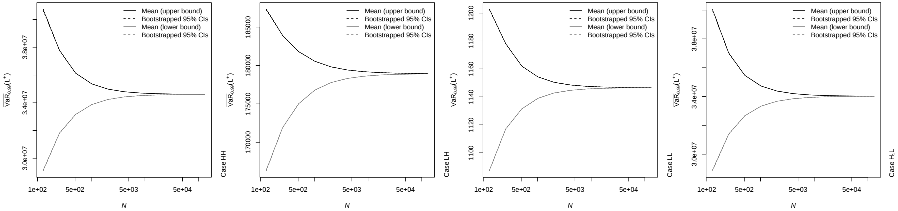
  Case LH : $\theta_1, \dots, \theta_d$ equidistant in $[1.5,\ 0.5]$ (light- to heavy-tailed)

  Case LL : $\theta_1, \dots, \theta_d$ equidistant in $[1.6,\ 1.4]$ (all light-tailed)

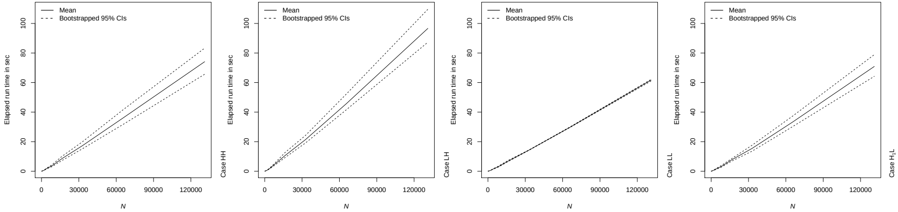
  Case H_1L: $\theta_2, \dots, \theta_d$ as in Case LL and $\theta_1 = 0.5$ (only first heavy-tailed)

- We consider $B = 200$ replicated simulation runs ($\Rightarrow$ empirical 95% confidence intervals); this allows us to study the effect of randomization.

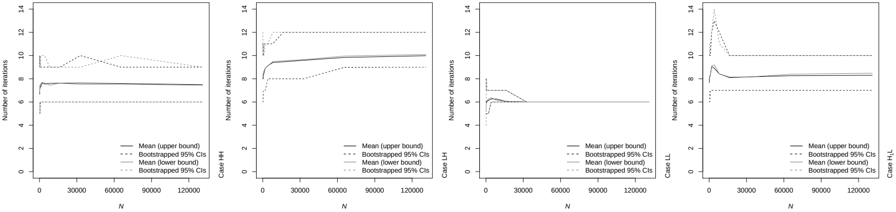
Results of Study 1 (N running, d fixed)



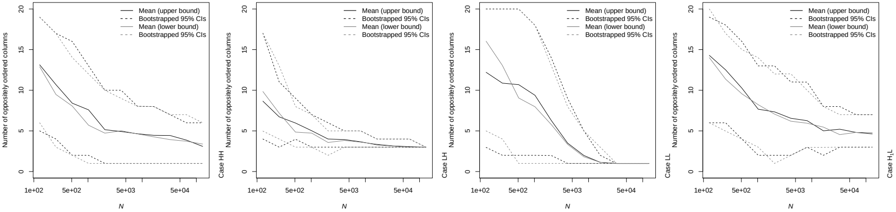
⇒ The means over all B computed $\underline{s}_N$ and $\overline{s}_N$ converge as N increases.



⇒ As N increases, run time (in s) increases ($\approx$ linearly).



⇒ The number of iterations rarely exceeds 12 as N increases.



⇒ The rate of decrease (# of opp. ordered columns) depends on the F_j's (especially small for Case LL); $\varepsilon =$ `NULL` not useful

3 The Adaptive Rearrangement Algorithm

- Algorithmically improved RA for computing $\underline{s}_N$ and $\overline{s}_N$; see `ARA()`.
- Improvements:

  1) Chooses more meaningful relative tolerances (and two!)

  2) Adaptively chooses N

3.1 How the ARA works

Algorithm (ARA for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$)

1) Fix $\alpha \in (0,1)$, $F_1^-, \dots, F_d^-$, a vector $\boldsymbol{N}$ and relative tol. $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2)$.

2) For $N \in \boldsymbol{N}$, do:

   2.1) Compute the lower bound $\underline{s}_N$:

   2.1.1) Define the (N,d)-matrix $\underline{X}^\alpha = \Big(F_j^-\big(\alpha + \frac{(1-\alpha)(i-1)}{N}\big)\Big)$.

   2.1.2) Randomly permute each column of $\underline{X}^\alpha$.

2.1.3) Iterate over each column of $\underline{X}^\alpha$ so that it becomes oppositely ordered to the sum of all others $\Rightarrow$ Matrix $\underline{Y}^\alpha$.

2.1.4) Repeat Step 2.1.3) until $\left|\frac{s(\underline{Y}^\alpha)-s(\underline{X}^\alpha)}{s(\underline{X}^\alpha)}\right| \le \varepsilon_1$ or until `maxiter` is reached. Then set $\underline{s}_N = s(\underline{Y}^\alpha)$.

2.2) Compute the upper bound $\overline{s}_N$: Similarly as in Step 2.1), but based on $\overline{X}^\alpha = \Big(F_j^-(\alpha + \frac{(1-\alpha)i}{N})\Big)$, compute $\overline{s}_N = s(\overline{Y}^\alpha)$.

2.3) If both ε_1 tolerances hold and $\left|\frac{\overline{s}_N - \underline{s}_N}{\overline{s}_N}\right| \le \varepsilon_2$, break.

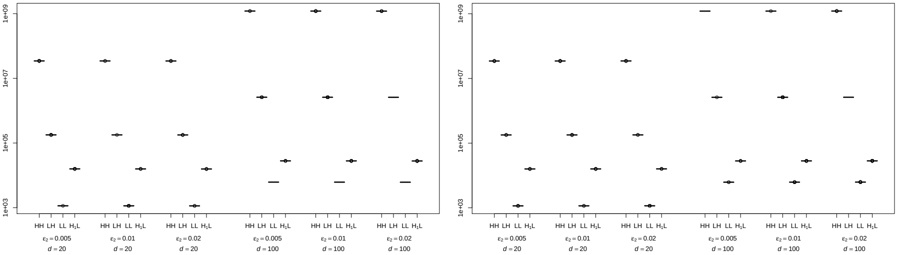
3) Return $(\underline{s}_N,\ \overline{s}_N)$ (rearrangement range; taken as $\overline{\mathrm{VaR}}_\alpha(L^+)$ bounds)

- If $\boldsymbol{N} = (N)$, the ARA reduces to the RA but uses *relative* individual *tolerances and joint convergence* is checked.
- Defaults (from simulations): $\boldsymbol{N} = (2^8, 2^9, \dots, 2^{20})$, `maxiter` $= 12$
- A useful choice for $\boldsymbol{\varepsilon}$ may be $\boldsymbol{\varepsilon} = (0.001,\ 0.01)$; can be freely chosen in `ARA()`.

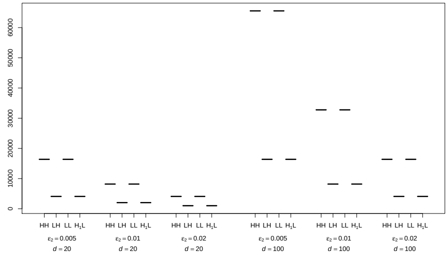
3.2 Empirical performance under various scenarios

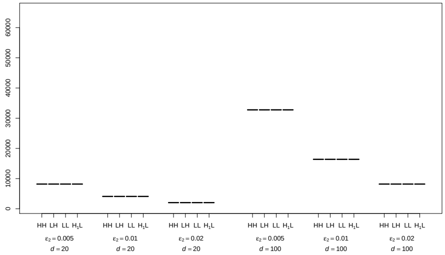
- As before: $d \in \{20, 100\}$, the Cases HH, LH, LL, $\mathrm{H_1L}$ and $B = 200$
- $\boldsymbol{\varepsilon} = (\varepsilon_1 = 0.1\%,\ \varepsilon_2 \in \{0.5\%, 1\%, 2\%\})$
- We investigate 1) $\underline{s}_N$, $\overline{s}_N$; 2) the N used in the final iteration; 3) the run time (in s); 4) the number of oppositely ordered columns; and 5) the number of iterations over all columns (for the last N used).

Boxplots of the $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left) and $\overline{s}_N$ (right):

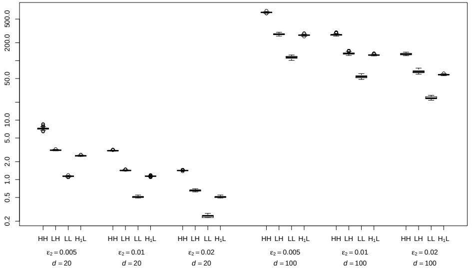


$\Rightarrow$ CIs are close; $\underline{s}_N$, $\overline{s}_N$ also close (as expected).

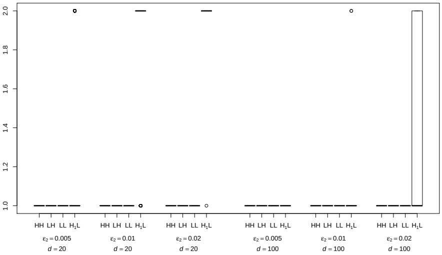


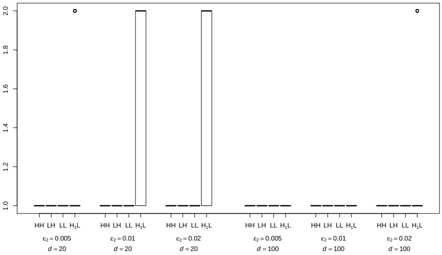


$\Rightarrow$ The N used differs for $\underline{s}_N$ (left) and $\overline{s}_N$ (right); but small for both.

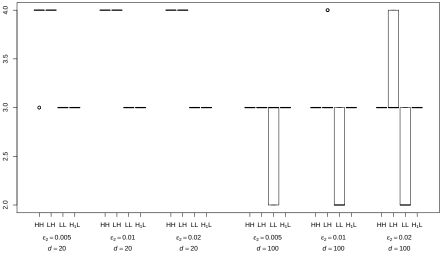


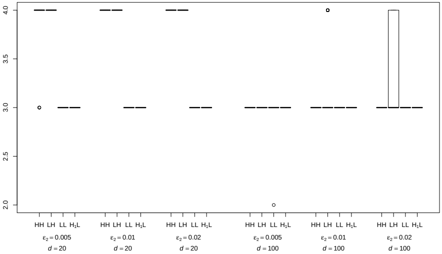
$\Rightarrow$ Doubling ε_2 reduces run time by $\approx 50\%$; good choice of ε_2 is important.





⇒ Only 1 or 2 are oppositely ordered (not worth spending more time...).





⇒ The number of iterations consistently remains below 5 (over all B runs).

Outlook

- DCARA (Dimension Reduction Adaptive Rearrangement Algorithm)
- DRARA (Divide and Conquer Adaptive Rearrangement Algorithm)
- How to use the reordering from the last N used before doubling N?
- How to apply the (A)RA without fitting the margins if the columns have different lengths?
- How to incorporate some information about the underlying copula C?
- Fast C/C++ version

References

Bernard, C. and McLeish, D. (2015), Algorithms for Finding Copulas Minimizing Convex Functions of Sums, http://arxiv.org/abs/1502.02130 (2015-02-24).

Bernard, C., Rüschendorf, L., and Vanduffel, S. (2013), Value-at-Risk bounds with variance constraints, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2342068 (2015-02-20).

Bernard, C., Denuit, M., and Vanduffel, S. (2014), Measuring Portfolio Risk under Partial Dependence Information, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2406377 (2015-02-20).

Embrechts, P., Puccetti, G., and Rüschendorf, L. (2013), Model uncertainty and VaR aggregation, *Journal of Banking and Finance*, 37(8), 2750–2764.

Embrechts, P., Puccetti, G., Rüschendorf, L., Wang, R., and Beleraj, A. (2014), An academic response to Basel 3.5, *Risks*, 2(1), 25–48.

Haus, U.-U. (2014), Bounding stochastic dependence, complete mixability of matrices, and multidimensional bottleneck assignment problems, *http://arxiv.org/abs/1407.6475*.

Jakobsons, E., Han, X., and Wang, R. (2015), General Convex Order on Risk Aggregation, *Scandinavian Actuarial Journal*.

Puccetti, G. and Rüschendorf, L. (2012), Computation of sharp bounds on the distribution of a function of dependent risks, *Journal of Computational and Applied Mathematics*, 236(7), 1833–1840.

Puccetti, G. and Rüschendorf, L. (2013), Sharp bounds for sums of dependent risks, *Journal of Applied Probability*, 50(1), 42–53.

Wang, R., Peng, L., and Yang, J. (2013), Bounds for the sum of dependent risks and worst Value-at-Risk with monotone marginal densities, *Finance and Stochastics*, 17(2), 395–417.