Computational aspects of quantitative risk management

Marius Hofert

2013-09-13
An example from QRM

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

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

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

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

**Goal:** Compute $\text{VaR}_\alpha(L_{t+1})$ with Monte Carlo (under certain assumptions).
exp(x) − 1 = 0 for x ≠ 0

Cancellation when |x| ≪ 1 due to subtracting floating-point numbers

How can this be tackled? High precision?
- Newton step for solving \( \log(1 + y) = x \) w.r.t. \( y \)

- Series representation:
  \[
  \exp(x) - 1 = x + \frac{x^2}{2} + \ldots
  \]
  (quadratic approximation)

- Near 0: \( \exp(x) - 1 \approx x \)
  (linear approximation)

Although this is just a (simple) example...
...we have already learned some important lessons:

- There might be computational problems lurking in the background.
- Between mathematics and applications, there is a computer ...and someone using it! (OpRisk: “risk of...loss resulting from inadequate or failed...people and systems...”)
- Mathematics: domain, limit
  Computer: domain, limit, and numerically critical region
- Properly implementing a model means taking care of this region.
- Multi-precision might be an option, but is slow(er) and less portable. Double-precision often suffices (when applied carefully).

Where did we encounter such problems? [we: Martin, Alex, me]
Density of a Gumbel copula

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

General formula for $c$ (trivial):

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

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

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

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

$$a_{dk}^G(1/\theta) = (-1)^{d-k} \sum_{j=k}^{d} \theta^{-j} s(d, j) S(j, k) = \frac{d!}{k!} \sum_{j=1}^{k} \binom{k}{j} \binom{j/\theta}{d} (-1)^{d-j}$$
Computational problem: Computing $(-1)^d \psi^{(d)}(t)$, then $\log(\cdot)$ fails!

⇒ Implemented 16 methods in \texttt{copula} (default depends on $t$, $d$, $\theta$).
One Idea: **exp-log-trick**

\[
\log P(x) = \log \sum_{k=1}^{d} a_{dk}^G (1/\theta) x^k
\]

\[
= \log \sum_{k=1}^{d} \exp(b_k), \quad b_k = \log(a_{dk}^G (1/\theta) x^k)
\]

\[
= \log \left( \exp(b_{\text{max}}) \cdot \sum_{k=1}^{d} \exp(b_k - b_{\text{max}}) \right)
\]

\[
= b_{\text{max}} + \log \sum_{k=1}^{d} \exp(b_k - b_{\text{max}})
\]

**Advantages:**

1) summands are in \((0, 1]\)

2) we can work in **log-scale**

3) similar trick for \(\log a_{dk}^G (1/\theta)\) (more difficult though)
Software problem: How to check results for correctness? CASs fail!
Example: \( \psi^{(50)}(15) = ? \) \( (\theta = 5/4; \text{correct answer: 1056.94}) \)
- Maple 14: 10 628, -29 800, . . . (chaotic; sign wrong; slow)
- Mathematica 8: – (aborted after 10 min)
- MATLAB 7.11.0: ✓ \( (d = 100: \text{aborted after several min}) \)
- 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 \( n, d, \theta \) . . .

\( \Rightarrow \) Parallel computing
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*)
Original goal: Computing \( \text{VaR}_\alpha \) with Monte Carlo

Recall:

\[
L_{t+1} = - \sum_{j=1}^{d} w_{t,j} (\exp(X_{t+1,j}) - 1)
\]

Scientific problem: Compute \( \text{VaR}_\alpha(L_{t+1}) \) with Monte Carlo and investigate it (in \( \alpha, d, n, C \) family, \( \tau \)).

Assumptions:

- For simplicity: \( w_{t,j} = \beta_j S_{t,j} = 1 \) (“frozen”)
- \( X = (X_{t+1,1}, \ldots, X_{t+1,d}) \sim H \), with
  \[
  H(x) = C(F_1(x_1), \ldots, F_d(x_d)),
  \]
  where \( F_1, \ldots, F_d \) are \( \mathcal{N}(0,1) \).

Translating this...
### to R:

<table>
<thead>
<tr>
<th>Variable</th>
<th>expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n.sim</td>
<td>$N_{sim}$</td>
<td>N</td>
<td>32</td>
</tr>
<tr>
<td>n</td>
<td>$n$</td>
<td>grid</td>
<td>64, 256</td>
</tr>
<tr>
<td>d</td>
<td>$d$</td>
<td>grid</td>
<td>5, 20, 100, 500</td>
</tr>
<tr>
<td>varWgts</td>
<td>$w$</td>
<td>frozen</td>
<td>1, 1, 1, 1</td>
</tr>
<tr>
<td>family</td>
<td>$C$</td>
<td>grid</td>
<td>Clayton, Gumbel</td>
</tr>
<tr>
<td>tau</td>
<td>$\tau$</td>
<td>grid</td>
<td>0.25, 0.50</td>
</tr>
<tr>
<td>alpha</td>
<td>$\alpha$</td>
<td>inner</td>
<td>0.950, 0.990, 0.999</td>
</tr>
</tbody>
</table>

```r
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)), # dimensions and weights
  varWgts = list(type="frozen", expr = quote(bold(w)),
                  value = list("5"=1, "20"=1, "100"=1, "500"=1)),
  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)

```r
# define the function doOne
doOne ← function(n, d, 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
  simRFC ← function(n, d, 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
    qnorm(rCopula(n, cop)) # simulate via Sklar's Theorem
  }
  X ← simRFC(n, d, family, tau) # simulate risk-factor changes

  ## 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
}
```

© 2013 Marius Hofert | RiskLab, ETH Zurich
The magic:

dolapply(), ..., doMclapply(), doClusterApply()

subjob()

docallWE()

doOne()

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):

```r
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:

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

<table>
<thead>
<tr>
<th></th>
<th>tau 0.25</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>256</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>family</th>
<th>tau 0.25</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>family</th>
<th>tau 0.25</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(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 = "...")
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0.25</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>95%</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d</td>
<td>α</td>
</tr>
<tr>
<td>τ</td>
<td></td>
<td>C</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clayton</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gumbel</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>500</td>
</tr>
</tbody>
</table>

© 2013 Marius Hofert | RiskLab, ETH Zurich
- `mayplot()`
- more visible than with tables
- new theoretical insights
- a lot more “behind the scenes”
Summary

- **Computational challenges** arise quickly . . .
  - . . . in **high dimensional** models (numerical vs theoretical)
  - . . . in **large-scale** studies (e.g., convergence of bootstraps)
  - . . . in **realistic** models (often not tractable anymore)
- To carefully **address computational issues** is one aspect . . .
- . . . another is that **model building should already take into account** computational aspects (not only mathematical beauty).
- Those **inventing new models** are often not those **implementing** them.
- **Provide running code** to demonstrate that your model works.
  - . . . also in non-typical/extreme situations (test it!).

You are in good company . . .
number of atoms in the universe

Atoms in the Universe - Universe Today
Jul 30, 2009 - The number of atoms in the entire observable universe is estimated to be within the range of 1078 to 1082. We’ve added the word ‘observable’...

Observable universe - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Observable_universe
Jump to Matter content - number of atoms - [edit source | edit]. Assuming the mass of ordinary matter is about 1.6×1053 kg (reference previous section) and ... Cosmological principle - Comoving distance - Recombination

Shannon number - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Shannon_number
As a comparison, the number of atoms in the observable universe, to which it is often compared, is estimated to be between 4×1079 and 1081.

How many atoms in the universe? - Yahoo! UK & Ireland Answers
uk.answers.yahoo.com › ... › Astronomy & Space
Aug 3, 2007 - The universe may perhaps be infinite, but we have no way of knowing if that is so, and so we can only sensibly discuss “the observable universe”.

Observable universe

In Big Bang cosmology, the observable universe consists of the galaxies and other matter that can, in principle, be observed from Earth in the present day—because light from those objects has had time ... Wikipedia

Feedback / More info
Thank you for your attention
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Loading required package: grDevices
> options(STERM='iESS', str.dendrogram.last="", editor='emacsclient', show.error.locations=TRUE)
> set.seed(1); rgamma(10, shape=1/1000) == 0
[1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE
- **Striking result:** \( \text{MSE} \propto \frac{1}{nd} \) (known margins)

- To show this theoretically:
  \[
  \sqrt{n} I(\theta_0)^{1/2} (\hat{\theta}_n - \theta_0) \xrightarrow{d} N(0, I_p)
  \]
  \[
  \Rightarrow \quad \hat{\theta}_n - \theta_0 \sim N(0, I(\theta_0)^{-1}/n)
  \]
  \[
  \Rightarrow \quad \mathbb{E}[(\hat{\theta}_n - \theta_0)^2] = \text{Var}(\hat{\theta}_n - \theta_0) = I(\theta_0)^{-1}/n
  \]

Open problem: \( I(\theta_0) \) linear in \( d \)?
Graphical GoF in higher dimensions

Pairwise Rosenblatt transformed observations to test $H^S_0$: $C$ is nested Gumbel with $\tau_0 = 0.2$, $\tau_1 = 0.4$, $\tau_2 = 0.4$

- 1000 simulations of $C_0(C_1(\cdot, \cdot), C_2(\cdot, \cdot, \cdot))$, (nested Gumbel) with $\tau_0 = 0.2$, $\tau_1 = 0.4$, $\tau_2 = 0.6$
- Panel plots customizable (Q-Q, etc.)
- See demo("gof_graph")

pp-values: minimum: 0.0019; global (Bonferroni/Holm): 0.039
Pairwise Rosenblatt transformed pseudo-observations
to test $H_0: C$ is $t_{11.96}$

- SMI log-returns
- 2011-09-09–2012-03-28 (141d)
- MLE for $\nu$
  (pairw. tau for $P$)

p-values: minimum: 0.084; global (Bonferroni/Holm): 1
The risk of applying a function incorrectly

R package mvtnorm (version \(\leq 0.9-9995\))

- Sampling \(N_d(\mu, \Sigma)\):
  \[
  \text{rmvnorm}(n, \text{mean}=\mu, \text{sigma} = \Sigma) \quad \checkmark
  \]

- Sampling \(t_\nu(\mu, \Sigma)\):
  \[
  \text{rmvt}(n, \text{mean}=\mu, \text{sigma} = \Sigma, \text{df} = \nu) \quad \times
  \]

**Problem 1:** \(\text{rmvt}()\) has no argument mean!
\[
\Rightarrow \text{normal mean-variance mixture } X = \sqrt{W} \mu + \sqrt{W} A Z
\]

**Problem 2:** \(\mu + \text{rmvt}(n, \sigma = \Sigma, df=nu) \quad \times
\]

**Problem 3:** documentation wrong
\[
(\Sigma \text{ is the dispersion matrix, not "covariance matrix"!})
\]

**Problem 4:** \(\nu = 0\) corresponds to \(N_d(\mu, \Sigma) \quad \times
\]

\[
\Rightarrow \text{Fixes in R-Forge version (CRAN } \geq 0.9-9996)\]
References

Hofert, M. (2013), On fallacies when sampling the multivariate $t$ distribution.

