Heavy-Tailed Distributions with Package `fitHeavyTail`

Prof. Daniel P. Palomar

Hong Kong University of Science and Technology (HKUST)

R/Finance 2023
University of Illinois at Chicago (UIC), Chicago, IL, USA
19-20 May 2023

Package `fitHeavyTail`

- R package `fitHeavyTail`: Mean and Covariance Matrix Estimation under Heavy Tails

  CRAN 0.2.0 | downloads 439/month | downloads 21K

- History:
  - Created in 2017.
  - First published in CRAN in 2019.
  - Latest version in 2023.
- A must for portfolio optimization with multiple assets.
- Authors and collaborators: Rui Zhou, Xiwen Wang, Esa Ollila, Frederic Pascal, and Daniel Palomar





Portfolio optimization needs $\boldsymbol{\Sigma}$

- Portfolio formulations typically require the $N \times N$ **covariance matrix** $\boldsymbol{\Sigma}$ of the N assets.
- There are many different formulations to optimize the **portfolio** vector $\mathbf{w}$.
- Markowitz proposed in his seminar 1952 paper (Markowitz, 1952) a trade-off between the portfolio expected return and its risk measured by the variance $\mathbf{w}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}$:

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{maximize}} & \mathbf{w}^\mathsf{T}\boldsymbol{\mu} - \lambda \mathbf{w}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w} \\ \text{subject to} & \mathbf{w} \in \mathcal{W}, \end{array}$$

  where $\mathcal{W}$ denotes some portfolio constraints such as $\mathbf{w} \geq \mathbf{0}$ and $\mathbf{1}^\mathsf{T}\mathbf{w} = 1$.
- Another popular formulation is the risk parity portfolio:[1]

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{minimize}} & \sum_{i,j=1}^{N} \left(w_i(\boldsymbol{\Sigma}\mathbf{w})_i - w_j(\boldsymbol{\Sigma}\mathbf{w})_j\right)^2 \\ \text{subject to} & \mathbf{w} \in \mathcal{W}. \end{array}$$

[1] R package `riskParityPortfolio` and Python package `riskparity.py`.

We could use the sample covariance matrix

- **Sample covariance matrix**:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{T-1}\sum_{t=1}^{T}(\mathbf{x}_t - \hat{\boldsymbol{\mu}})(\mathbf{x}_t - \hat{\boldsymbol{\mu}})^{\mathsf{T}}.$$

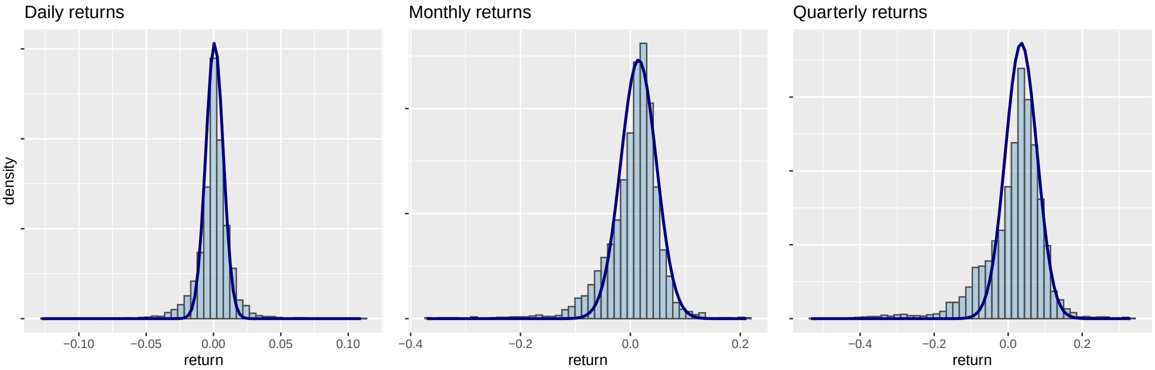
  where $\mathbf{x}_t$ contains the returns of the N assets at period t and $\hat{\boldsymbol{\mu}}$ is an estimation of the mean vector.

- Good properties of the sample covariance matrix:
  - **unbiased** estimator: $\mathrm{E}[\hat{\boldsymbol{\Sigma}}] = \boldsymbol{\Sigma}$
  - **consistent** estimator: $\lim_{T\rightarrow\infty} \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}$
  - **optimal** from the perspective of Gaussian maximum likelihood estimation:

$$\underset{\boldsymbol{\mu},\boldsymbol{\Sigma}}{\text{minimize}} \quad \log\det(\boldsymbol{\Sigma}) + \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_t - \boldsymbol{\mu}).$$

But financial data is not Gaussian

- One of the well-known stylized facts of financial data is the heavy-tailed distribution (not Gaussian).
- Histogram of S&P 500 returns at different frequencies (with Gaussian fit):



We really need a heavy-tailed estimator

- We cannot ignore the need for a proper estimator under heavy tails.
- Maximum likelihood estimator under the Student's t distribution:

$$
\begin{aligned}
\underset{\boldsymbol{\mu},\boldsymbol{\Sigma},\nu}{\text{minimize}} \quad & \log\det(\boldsymbol{\Sigma}) + \frac{\nu+N}{T}\sum_{t=1}^{T}\log\left(1+\frac{1}{\nu}(\mathbf{x}_t-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_t-\boldsymbol{\mu})\right)\\
& +2\log\Gamma(\nu/2)+N\log(\nu)-2\log\Gamma\left(\frac{\nu+N}{2}\right).
\end{aligned}
$$

- But it does not have a closed-form solution.

Iterative algorithms

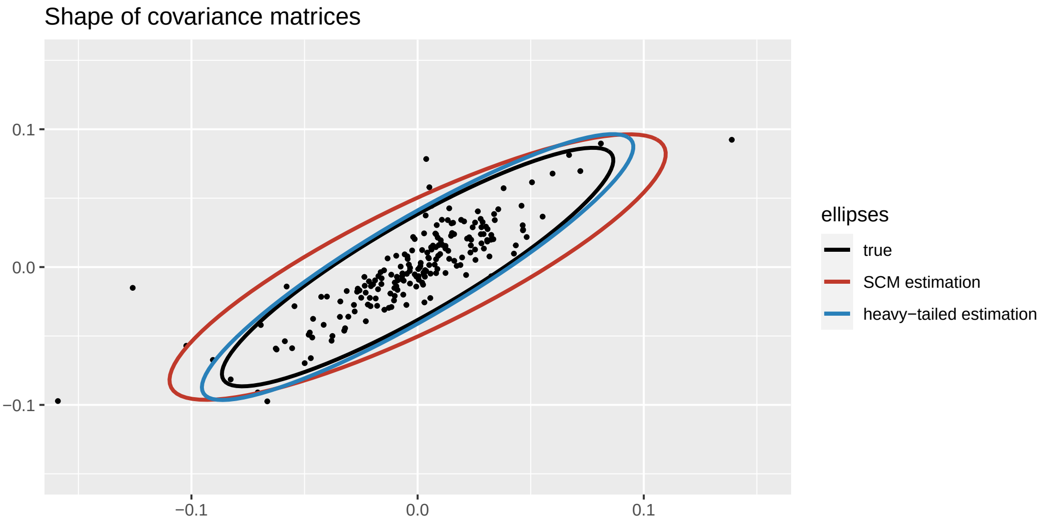
- The solution is in the form of fixed-point equations:

$$\boldsymbol{\mu} = \frac{\frac{1}{T}\sum_{t=1}^{T} w_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \times \mathbf{x}_t}{\frac{1}{T}\sum_{t=1}^{T} w_t(\boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{t=1}^{T} w_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \times (\mathbf{x}_t - \boldsymbol{\mu})(\mathbf{x}_t - \boldsymbol{\mu})^{\mathsf{T}},$$

  where $w_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\nu + N}{\nu + (\mathbf{x}_t - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_t - \boldsymbol{\mu})}$.
- Iterative algorithms to compute these fixed-point equations based on the majorization-minimization (MM) framework (Sun et al., 2017) are implemented in the package `fitHeavyTail`:
  - function `fit_Tyler()`: (Sun et al., 2014)
  - function `fit_Cauchy()`: (Sun et al., 2015)
  - function `fit_mvt()`: (Liu and Rubin, 1995; Liu et al., 1998), (Ollila et al., 2021), and (Pascal et al., 2021)
  - function `fit_mvst()`: (Aas and Haff, 2006)

Gaussian versus heavy-tailed estimation



How to use package fitHeavyTail

- Generate synthetic data:

```
library(mvtnorm)  # package for multivariate t distribution

N <- 10   # number of variables
T <- 80   # number of observations
nu <- 4   # degrees of freedom for heavy tails
mu <- rep(0, N)
# create covariance matrix with factor model structure
set.seed(42)
U <- t(rmvnorm(n = round(0.3*N), sigma = 0.1*diag(N)))
Sigma_cov <- U %*% t(U) + diag(N)
Sigma_scatter <- (nu-2)/nu * Sigma_cov

# generate TxN data matrix
X <- rmvt(n = T, delta = mu, sigma = Sigma_scatter, df = nu)
```

How to use package fitHeavyTail

- Tradidional sample estimators:

```
mu_sm     <- colMeans(X)
Sigma_scm <- cov(X)
```

- Heavy-tailed estimators:

```
library(fitHeavyTail)
fitted <- fit_mvt(X)
```

```
names(fitted)
```

```
R>> [1] "mu"          "scatter"       "nu"               "mean"
R>> [5] "cov"         "converged"     "num_iterations" "cpu_time"
```

How to use package fitHeavyTail

- Compare errors in $\boldsymbol{\mu}$ (half!):

```
sum((mu_sm      - mu)^2)
sum((fitted$mu - mu)^2)
```

```
R>> [1] 0.2857323
R>> [1] 0.1487845
```

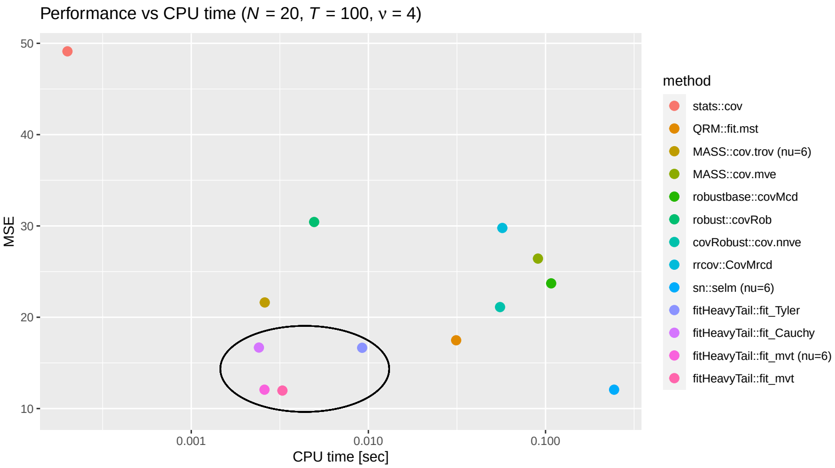
- Compare errors in $\boldsymbol{\Sigma}$ (half!):

```
sum((Sigma_scm  - Sigma_cov)^2)
sum((fitted$cov - Sigma_cov)^2)
```

```
R>> [1] 5.861138
R>> [1] 3.031499
```

Comparison with other packages

- The methods provided by the package `fitHeavyTail` (see the ellipse) clearly offer the best trade-off of estimation error and CPU time.



Thanks

For more information visit:

https://www.danielppalomar.com

https://github.com/dppalomar

https://www.youtube.com/danielpalomar

References I

Aas, K., and Haff, I. H. (2006). The generalized hyperbolic skew student'st-distribution. *Journal of Financial Econometrics*, *4*(2), 275–309.

Liu, C., and Rubin, D. B. (1995). ML estimation of the t-distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, *5*(1), 19–39.

Liu, C., Rubin, D. B., and Ying Nian Wu, and. (1998). Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika*, *85*(4), 755–770.

Markowitz, H. (1952). Portfolio selection. *J. Financ.*, *7*(1), 77–91.

Ollila, E., Palomar, D. P., and Pascal, F. (2021). Shrinking the eigenvalues of M-estimators of covariance matrix. *IEEE Transactions on Signal Processing*, *69*, 256–269.

Pascal, F., Ollila, E., and Palomar, D. P. (2021). Improved estimation of the degree of freedom parameter of multivariate t-distribution. In *Proc. European signal processing conference (EUSIPCO)*. Dublin, Ireland.

Sun, Y., Babu, P., and Palomar, D. P. (2014). Regularized Tyler's scatter estimator: Existence, uniqueness, and algorithms. *IEEE Trans. Signal Processing*, *62*(19), 5143–5156.

References II

Sun, Y., Babu, P., and Palomar, D. P. (2015). Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions. *IEEE Trans. Signal Processing*, *63*(12), 3096–3109.

Sun, Y., Babu, P., and Palomar, D. P. (2017). Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, *65*(3), 794–816.