Portfolio Optimization using Second-order Cone Programming in the NAG Library

Shuanghua Bai & Charlotte Silvester (Numerical Algorithms Group)

Update to the NAG SOCP solver

- In Mark 29.3 of the NAG Library, we have updated the second-order cone programming (SOCP) solver to increase the performance.
- This solver has proved to be especially effective for dense quadratic problems, widely used in portfolio optimization.
- The NAG solver is fully integrated in CVXPY in Python, try out this functionality via the QR code.

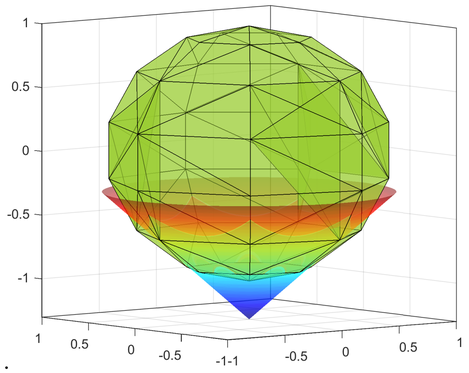
What is SOCP?

SOCP is convex optimization which extends linear programming (LP) with second-order (Lorentz or the ice cream) cones. It appears in a **broad range of applications** from **quantitative finance** to quadratic programming and **robust optimization**. It has become an important tool for financial optimization due to its **powerful nature**. Interior point methods (IPM) are the most popular approaches to solve SOCP problems due to their theoretical polynomial complexity and **practical performance**. The NAG SOCP solver (e04pt) uses an IPM to solve a problem in the standard form:

$$\begin{aligned} \underset{x\in\Re^n}{\text{minimize}} \;& c^T x \\ \text{subject to} \;& l_A \le Ax \le u_A, \\ & l_x \le x \le u_x, \\ & x \in \mathcal{K}, \end{aligned}$$

where $A \in \Re^{m\times n}$, $l_A, u_A \in \Re^m$, $c, l_x, u_x \in \Re^n$ are the problem data, and $\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r} \times \Re^{n_l}$ where $\mathcal{K}^{n_i}$ is a second-order cone defined as

$$\mathcal{K}_q^{n_i} := \left\{ x = (x_1, \ldots, x_{n_i}) \in \Re^{n_i} \;:\; x_1^2 \ge \sum_{j=2}^{n_i} x_j^2,\; x_1 \ge 0 \right\}.$$



Feasible region of an SOCP problem with 3 variables.

Usage of SOCP in Portfolio Optimization

SOCP is widely used in quantitative finance due to its **flexibility and versatility** to handle a large variety of problems with different kinds of constraints, such as stochastic and robust portfolio optimization. Here are some portfolio optimization problems and constraints that can be **handled by SOCP**.

- Maximize Sharpe ratio:

$$\max \frac{\mu^T x}{\sqrt{x^T \Sigma x}}.$$

- Tracking error constraint:

$$\frac{1}{2} x^T P x + q^T x + r \le 0.$$

- Leverage constraint:

$$\|x\|_1 \le L.$$

- Holding and budget constraints:

$$0 \le x \le u, \quad e^T x = b,$$

where e is the vector of all ones.

- Maximum position constraint:

$$\max |x| \le m.$$

- Market impact cost:

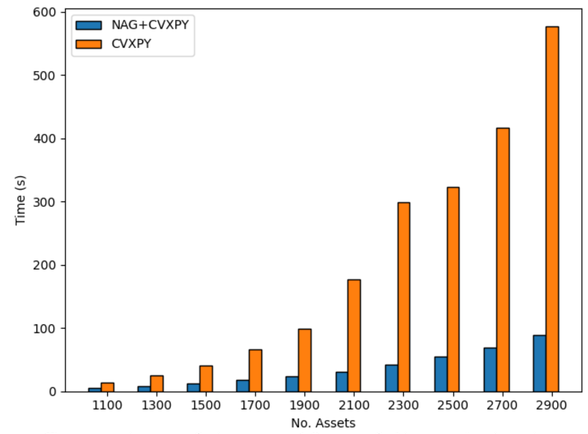
$$w^T |x - x_0|^\eta \le u_M.$$

Performance of the NAG SOCP solver

We compare the solve times of the NAG SOCP solver called from CVXPY to the default solver in CVXPY on 10 classic portfolio optimization problems (ranging from 1100 to 2900 assets). Randomly generated data was used in a Markowitz model whose objective involved a large covariance matrix and was subject to a long-only constraint and a budget constraint. (Default options, accuracy 10^{-8}, single-threaded mode).

- For smaller problems, the NAG solver is at least **three times faster** than standard CVXPY.
- For larger problems, the NAG solver becomes as much as **six times faster** than standard CVXPY.
- The NAG solver achieves the above performance while maintaining a high level of **accuracy**.

The NAG SOCP solver is highly performant on portfolio optimization problems.

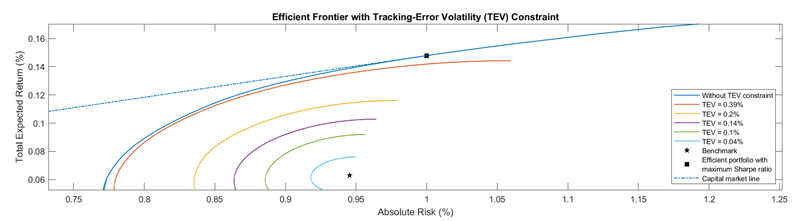


Comparison of time on 10 portfolio optimization problems.

Case study: Portfolio optimization with tracking-error constraints

The following model explores the risk and return relationship of **active portfolios** subject to **tracking-error volatility (TEV)**. Let r be the vector of expected returns and V the covariance matrix for asset returns (estimated from daily data for 30 stocks in DJIA from March 2018 to March 2019). Randomly generating a benchmark portfolio b, we solve the following optimization:

$$\begin{aligned} \underset{x\in\Re^{30}}{\text{minimize}} \;& -r^T x + \mu(b+x)^T V (b+x) \\ \text{subject to} \;& e^T x = 0, \\ & x + b \ge 0, \\ & x^T V x \le tev, \end{aligned}$$



where e is the vector of all ones, μ controls the trade-off between excess return and absolute risk and tev is the threshold on TEV. Note that without the absolute risk in the objective, the problem reduces to excess-return optimization. However, *Roll (1992)* noted this *classical model leads to the unpalatable result that the active portfolio has systematically higher risk than the benchmark* and is not optimal. Therefore, by taking the absolute risk into account the **QCQP model (solved by SOCP) improves the performance of active portfolio**.

SOCP solver in the NAG Library was used to solve the above model. Each efficient frontier in the figure was generated by solving 2000 SOCPs. The whole process took around 4 mins (less than 0.02s per solving).

Various financial models can be significantly enhanced by using the NAG SOCP solver.