Portfolio Optimization with Trading Cost

Andrey Kuzmenko
for.akuz@gmail.com
http://akuz.me

Mar 9, 2014

1 Introduction

Let us consider mean-variance portfolio optimization problem stated as:

$$max_w \;\; \mu^T w - \frac{\gamma}{2} w^T C\, w$$

where γ is a risk aversion coefficient. This problem has an explicit solution, which can be derived using Lagrange multipliers:

$$w_1 = \frac{1}{\gamma} C^{-1} \mu$$

2 Trading Cost

Let us now consider mean-variance portfolio optimization with trading cost:

$$max_w \;\; \mu^T w - \frac{\gamma}{2} w^T C\, w - c\, \mathbf{1}^T \left| w - w_0 \right|$$

Since absolute value is a non-differentiable function, we cannot solve this problem using Lagrange multipliers. Let us now reformulate this problem by splitting the trades $w - w_0$ into the buy and sell components. We will then be able to arrive at the general quadratic optimization problem which can be solved using an efficient iterative method.

Let $w - w_0 = b - s$ with b, $s \geq 0$ being the buy and sell orders. We can now rewrite the porfolio optimization problem as follows:

$$\begin{array}{ll} max_{b,s\geq 0} & \mu^T \left(w_0 + b - s\right) \\ & -\frac{\gamma}{2} \left(w_0 + b - s\right)^T C\, \left(w_0 + b - s\right) \\ & -c\, \mathbf{1}^T \left(b + s\right) \end{array}$$

$$\begin{array}{ll} max_{b,s\geq 0} & \mu^T w_0 + \mu^T b - \mu^T s \\ & -\frac{\gamma}{2} \left(w_0^T C\, w_0 + w_0^T C\, b - w_0^T C\, s + b^T C\, w_0\right) \\ & -\frac{\gamma}{2} \left(b^T C\, b - b^T C\, s - s^T C\, w_0 - s^T C\, b + s^T C\, s\right) \\ & -c\, \mathbf{1}^T \left(b + s\right) \end{array}$$

$$\begin{array}{ll} max_{b,s\geq 0} & \mu^T b - \mu^T s \\ & -\frac{\gamma}{2}\left(2w_0^T C\, b - 2w_0^T C\, s\right) \\ & -\frac{\gamma}{2}\left(b^T C\, b - 2b^T C\, s + s^T C\, s\right) \\ & -c\,\mathbf{1}^T\,(b+s) \end{array}$$

$$\begin{array}{ll} max_{b,s\geq 0} & \left(\mu^T - \gamma w_0^T C - c\,\mathbf{1}^T\right) b \\ & +\left(-\mu^T + \gamma w_0^T C - c\,\mathbf{1}^T\right) s \\ & -\frac{\gamma}{2}\left(b^T C\, b - 2b^T C\, s + s^T C\, s\right) \end{array}$$

Or alternatively as a minimization problem:

$$\begin{array}{ll} min_{b,s\geq 0} & \left(-\mu^T + \gamma w_0^T C + c\,\mathbf{1}^T\right) b \\ & +\left(\mu^T - \gamma w_0^T C + c\,\mathbf{1}^T\right) s \\ & +\frac{\gamma}{2}\left(b^T C\, b - 2b^T C\, s + s^T C\, s\right) \end{array}$$

3 Full Problem

Let us now combine b and s into a single vector of orders:

$$x = \left[\begin{array}{c} b \\ s \end{array}\right]$$

We can now rewrite the minimization problem as:

$$min_{x\geq 0}\ \ \alpha^T x + \frac{\gamma}{2} x^T \Phi\, x$$

where:

$$\alpha = \left[\begin{array}{r} -\mu + \gamma C\, w_0 + c\,\mathbf{1} \\ \mu - \gamma C\, w_0 + c\,\mathbf{1} \end{array}\right]$$

$$\Phi = \left[\begin{array}{rr} C & -C \\ -C & C \end{array}\right]$$

This problem can now be solved using a quadratic optimization algorithm (for example, the interior point method).

4 Numerical Trick

The matrix Φ is positive semi-definite (because C is positive definite, and Φ is rank defficient). In ideal, Φ will have half of the eigenvalues equal to zero. However, due to the round-off error, these eigenvalues can be in fact calculated as small negative or positive values, which will prevent the application of the quadratic optimization algorithm, which requires a positive (semi-)definite quadratic form. A simple trick would be to make Φ positive definite by adding small positive values to its diagonal, and solving the modified optimization problem with $\Phi^{'} = \Phi + \varepsilon I$. Provided ε is small relative to the elements of C, the solution will not differ much from the solution of the unmodified problem.

5 Two Step Trick

The full optimization problem setup involves a quadratic form Φ of size $2K$, where K is the number of stocks. Algorithmic complexity of solving the quadratic optimization problem with a positive definite quadratic form is, in general, of polynomial time. Therefore, the problem with dimensionality of $2K$ might in general require much greater time than the time, which would be required to solve 2 problems with of dimensionality K. This is in fact confirmed in practice, with $2K$-sized optimization problem solution taking as much as $\times 8$ time required to solve K-sized optimization problem.

In order to speed up the optimization process, the following two step trick can be used. In the first step, we solve the unconstrained problem without cost. Using the solution from the first step, we can identify in which direction does the unconstrated optimization wants to change the positions. In the second step, we restrict position changes to the direction identified in the first step. Since we now know the direction of position changes, we can introduce the trading cost for each stock individually, setting it to positive value for stocks that we want to buy, and to negative value for stocks that we want to sell. Solving this second optimization problem, we identify the new positions, taking into account the trading cost.

This approach introduces approximation, since we first identify the direction of position changes without taking into account the trading cost. In practice, this has very little effect on the final solution compared to the solution of the full $2K$-sized problem, resulting in almost identical optimized positions and portfolio performance. However, this approach enables improving the algorithmic complexity of portfolio optimization by up to an order of magnitude, and thus respectively decreasing the backtest (and real-time portfolio optimization, if required) time by an order or magnitude.