Name: Shawn Feng

Date: 6/10/2019

Topic: Weekly report (6/2-6/9) – wrap up "osqp" optimization

In this week, I implemented "osqp" optimizer to "PortfolioAnalytics." "osqp" is a mathematical optimizer which can solve quadratic programming problems fast. In equation form: it can find $min\ w'Pw + q'x$ subject to $u < Aw < l$. However, it cannot take too many constraints and objectives since we need to transform all constraints and objectives into matrix format. Currently, it can find "max return", "min volatility" and "max Sharpe Ratio" under "group", "box" and "return target" constraints.

Before I started "osqp" engine, I set up the constraints matrices. First, I embedded the "group" constraints into the box constraints. The "group" constraints is a list in constraints parameter, I transformed it into matrix form and row-bind it with "box" constraints. Using "box" and "return target" constraints we can get the constraint-matrices: $A$, $u$ and $l$.

The "max return" and "min volatility" parts are easy to solve; I just set the $P$ or $q$ matrix to 0 matrix in quadratic programming equation. In this step, we can also get the upper return level and lower return level for the next step, if there is one. Also, if the users input a target return, the optimization will find the min volatility for that target return.

The "max Sharpe Ratio" algorithm of "osqp" has two parts. If users input a risk aversion parameter, we will use the modern portfolio theory to find the desired portfolio. Otherwise, the "osqp" will find the upper return level (max return) and lower return level (according to min volatility). From these two levels, I used a quartile recursion method to apply "osqp" solver to find the max Sharpe ratio (in this part we will update $A$, $u$ and $l$ in every recursion process).

I used two real financial data samples to do the comparison test; one is a CTA monthly return dataset with 12 dimensions and 5-years history, the other one is an ETF monthly return dataset with 37 dimensions and 5-years history. The test target is simple - find a portfolio with max Sharpe Ratio. The test constraints conclude "weight sum", "long only" and "group". The "group" constraint assigned four sub-pools with max and min sum-weight from the whole asset pool.

This is only a sample test, I also did variety tests with random box constraints. Results from random tests are consistence with this sample test.

The following chart is the result from comparison tests:

| ETF Test | | | | |
|---|---|---|---|---|
| | Mean (%) | Sigma (%) | Sharpe Ratio (%) | Running Time (s) |
| DEoptim | 0.35 | 1.45 | 24.05 | 69.0852 |
| random | 0.41 | 1.64 | 24.82 | 156.6147 |
| pso | 0.37 | 2.61 | 14.17 | 194.1641 |
| GenSA | 0.45 | 2.78 | 16.22 | 941.3827 |
| osqp | 0.93 | 2.59 | 35.97 | 0.028 |
| CTA Test | | | | |
| | Mean (%) | Sigma (%) | Sharpe Ratio (%) | Running Time (s) |
| DEoptim | 0.16 | 0.84 | 19.5 | 15.4168 |
| random | 0.14 | 0.94 | 14.77 | 78.7616 |
| pso | 0.21 | 1.25 | 16.52 | 30.8647 |
| GenSA | 0.2 | 1.49 | 13.71 | 59.8354 |
| osqp | 0.32 | 1.33 | 24.11 | 0.0156 |

In this chart, we can see that the "osqp" optimization only needs milliseconds to find the max Sharpe Ratio while other optimization methods require minutes to find a smaller Shape Ratio. We can say that the "osqp" is a competitive optimization on finding max Sharpe Ratio. I also attached weight detail of these two tests in the .csv file.

Besides the speed and accuracy test, I did a stable test. Using "osqp" on CTA problem set, I computed 100 times and got a standard error 0.00004. The standard error for the second fast method "DEoptim" is 0.0037. This means the result from "osqp" is stable enough that we can see its result is consistent.

In conclusion, if we have fewer constraints and want to find a max Sharpe Ratio, "osqp" can save us minutes even hours; meanwhile, it can provide a bigger and stabler Sharpe Ratio.