Addition of aqr_epo_2020

HangukQuant[1, 2*]

May 10, 2023

[1] [2]

Abstract

We add the Enhanced Portfolio Optimization (EPO, Pedersen et al. [1][3]) to the Russian Doll testing engine, alongside other methods previously added to the Python library. Here we give succinct, key summary of the intuition behind the EPO portfolio. The code is appended to the attendant post on hangukquant.com. The datasets are also found in the post.

0.1 A Note of Precaution

There is absolutely no warranty or guarantee implied with this product. Use at your own risk. I provide no guarantee that it will be functional, destructive or constructive in any sense of the word. Use at your own risk. Trading is a risky operation.

1 Review of Key Arguments

The detailed arguments and in depth analysis of the Enhanced Portfolio Optimization method is found in the paper referenced. We repeat the core arguments. Recall (in the absence of long-only constraints), the rational investor (with accompanying utility and normality assumptions) make

[1] ***1**: hangukquant@gmail.com, hangukquant.substack.com

[3] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3530390

the choice $\max_x(x's - \frac{\gamma}{2}x'\Sigma x)$, where x is the portfolio weight, s is the expected return and Σ is the covariance matrix of asset returns. The closed form solution to this is based on the first order condition and yields the mean variance optimal portfolio $x^* = \frac{1}{\gamma}\Sigma^{-1}s$. The (symmetric, semi-positive definite) covariance matrix has variance-correlation decomposition given by $\Sigma = \sigma\Omega\sigma$, where $\sigma = diag((\sqrt{\Sigma_{ii}})_{i\in[len(x)]})$, and Ω is the correlation matrix of asset returns. The correlation matrix is symmetric and hence orthogonally diagonalizable (see our linear algebra notes); it has eigenvector decomposition. Using principal component analysis, we map the portfolios onto unit variance space, where the set of principal component portfolios are orthogonal (more specifically, orthonormal). In particular, the first principal component portfolio is the weighting h that maximises the variance $h'\Omega h$ s.t. $h'h = 1$. The second principal component maximises $h'\Omega h$ subject to $h'h = 1$ and being orthogonal to the first principal component and so on. The proportion of variance explained by these principal component portfolios successively decrease. The eigen-decomposition can be written $\Omega = PDP'$ where $P' = P^{-1}$ and matrix P are columns of eigenvectors of orthogonal portfolios in the unit variance space (our principal components). The principal component portfolios have expected excess returns $s^p = P'\sigma^{-1}s$ and covariance matrix given by the diagonal D. Using the property that for diagonal matrix D, we have $D = D'$, and $D^p = (d^p_{ij})$, we may reformulate the investor utility function

$$x's - \frac{\gamma}{2}x'\Sigma x = (P'\sigma x)'s^p - \frac{\gamma}{2}(P'\sigma x)'D(P'\sigma x) = z's^p - \frac{\gamma}{2}z'Dz, \tag{1}$$

where $z = P'\sigma x$ are weighting for investments in the principal component portfolios. As in classical MVO the investor makes the investment $z^* = \frac{1}{\gamma}D^{-1}s^p$, and by orthogonality the portfolio-wise investment made is $z^*_i = \frac{1}{\gamma}\frac{s^p_i}{\sqrt{D_i}}\frac{1}{\sqrt{D_i}}$, a product of the i-th principal component portfolio sharpe and factor $\frac{1}{\sqrt{D_i}}$. This equation tells us that the estimation error in s^p_i is magnified by $\frac{1}{\sqrt{D_i}}$, the value of which is the largest in the least significant principal component portfolios. The low risk also bloats the estimation of Sharpe ratios of these portfolios. These portfolios are coined as the 'problem portfolios', and the EPO shrinkage targets this by shrinking the estimated variance of these principal component portfolios towards the average. Recall from the linear algebra notes that $tr(AB) = tr(BA)$. Then $tr(\Omega) = tr(PDP') = tr(DP'P) = tr(D\mathbb{1}) = tr(D)$. The shrinkage is simply towards identity since Ω is correlation matrix with diagonal ones; the shrinkage solution to the correlation matrix becomes

$$\tilde{\Omega} = P\tilde{D}P' = P((1-\theta)D + \theta\mathbb{1})P' = (1-\theta)\Omega + \theta\mathbb{1}, \tag{2}$$

with covariance matrix estimator $\sigma\tilde{\Omega}\sigma$ as input to the mean variance optimization problem. We implemented the shrinkage $\theta = 0.75$ in the code as mentioned in the paper, but it should be trivial for readers to extend this and run their own in-sample walkforward optimization for dynamic shrinkage values.

2 Notes

The updated Russian Doll Python library is found on the post. These can be applied to arbitrary strategies by specifying the configurations dictionary as input to the Amalgapha object. For instance, these configurations all specify different, valid instructions to the multi-strategy backtester for combining active alphas:

```
{"framework": "parity"},
{"framework": "parity","positional_inertia": 0.15},
{"framework": "quadratic_risk_lc","ret": {"est": "simple_mean"},"cov": {"est": "sample"}},
{"framework": "quadratic_risk_lc","ret": {"est": "simple_mean"},"cov": {"est": "ledwol_cc"}},
{"framework": "quadratic_risk_lc","ret": {"est": "simple_mean"},"cov": {"est": "aqr_epo_2020"}}
```

The datasets are found on the *mega.nz* link on the post. Once a dataset is posted in a newer post, the old links will be taken down. The *master.json* file contains the datasets and formulaic alphas that are used in the simulation. The entries are SHA256 hash values that point to a simulation dataframe created by the Russian Doll backtesting module. For readers who want to play around with these simulation data, you can download them from the *'simulation_data_numX.lz'* files. The simulation dataframe contains information such as portfolio holdings, weightings, pnl, execution costs (assumed net of 0.1% notional volume), cost-free returns, leverage applied and so on. The universe of assets (all US stocks) are easily recoverable from any of the simulation dataframes. Readers may obtain the pricing data from any of the open-source data libraries or commercial datasets. All lz files are Python pickles that can be loaded by running the code:

```
with lzma.open(path, 'rb') as fp:
    file = pickle.load(fp)
```

These are not compulsory, since the Russian Doll module is designed to work with arbitrary strategies. Readers should be able to pass in their own proprietary trading strategies and run the optimization code should they be so willing.

3 On Future Work

We make some comments in the post on the efficacy of these portfolio management methods. For the weeks to come, we will continue to review, discuss and implement some of these portfolio methods found in literature and integrate it into the Russian Doll. The takeaway should not be that any of these methods are 'the golden ticket' to portfolio management. Our goal for the reader is to appreciate these methods and the variable settings in which one may outperform the other, so that one may apply the correct method depending on the nature of their own problem domain. If you are convinced that there is a golden solution that is the answer to all of your portfolio management problems, the reader would be in for a ride of disappointment with us (see No Free Lunch Theorem).

References

[1] L. H. Pedersen, A. Babu, and A. Levine. Enhanced Portfolio Optimization. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3530390.