NUS INVESTING SOCIETY

Technical Report

BLOpt: An Open-Source Library for Black-Litterman Portfolio Optimization

Alberto Costa[1,2], Penghao Chen[2], Zhang Han[2], Elroy Haw[2], Kaustubh Jagtap[2], Zexin Wang[2], and Hao Wu[2]

[1]E-mail: costa@lix.polytechnique.fr

[1]*ETH Zurich, Future Cities Laboratory, Singapore*
[2]*National University of Singapore*

December 2017

National University of Singapore - Mochtar Riady Building
BIZ 1, #2-7, 15 Kent Ridge Drive
119245, Singapore

Abstract

The Black-Litterman model has been put forward as a methodology for overcoming some issues of classical portfolio optimization. We propose BLOpt, a Python open-source library which implements the main portfolio optimization methods, including Black-Litterman. Short-selling, transactions costs, and portfolio rebalancing have been considered as well. Computational results on the components of the Dow Jones Industrial Average are provided to show how the library works. In addition, black-box optimization is employed to derive a good vector of investor views for the Black-Litterman model. This allows checking, in a backtesting simulation, if those views would have been compatible with the relevant global macro indicators.

Keywords: Black-Litterman, black-box optimization, portfolio optimization.

Contents

1 Introduction

Modern Portfolio Theory is a mathematical framework for generating an optimal portfolio of assets according to some investing goal. Traditionally, the classical Markowitz portfolio optimization model is used to combine two objectives [8]:

- maximization of expected returns
- minimization of risk.

However, the Markowitz model has some drawbacks. First, it requires expected returns and covariance to be specified for each asset of interest. Since they are based on historical data, they may not be reliable to predict future performance. Second, it can lead to portfolios with extreme and unintuitive weights, which are not so appealing to investors and fund managers. In order to overcome these issues, the Black-Litterman model has been proposed in [3, 4].

The Black-Litterman model is a methodology that takes into account the subjective views of the investor regarding performance of one or more assets, thus providing an updated vector of expected returns that are more likely to provide portfolios in line with the investor's expectations.

This paper is organized as follows. In Section 2 the Modern Portfolio Theory and the Black-Litterman model are introduced. Those methods have been implemented in an open-source Python library described in Section 3. After that, some test cases to show how the library can be used are presented in Section 4. Finally, conclusions are reported in Section 5.

2 Background

In this section we introduce the basic methods of the Modern Portfolio Theory and the Black-Litterman formula that are implemented in the Python library.

As notation, vectors and matrices are in bold, x_i represents the i-th element of the vector $\boldsymbol{x}$, $\Sigma_{i,j}$ is the element in the i-th row and j-th column of matrix $\boldsymbol{\Sigma}$, and

$\boldsymbol{x}^T$ is the transpose of the vector $\boldsymbol{x}$ (similar notation holds for the transpose of a matrix). If R_i is a random variable, $\mu_i = E[R_i]$ represents its expected value.

2.1 Modern Portfolio Theory

Modern Portfolio Theory provides a framework to define how risk-averse investors can construct portfolios that maximize their expected return based on a given level of market risk. Conversely, given a desired level of expected return, an investor can construct a portfolio with the smallest possible risk. The latter portfolio, also known as the Global Minimum-Variance Portfolio (GMVP), can be found by solving the following quadratic optimization problem:

$$\min \quad \sum_{i \in N} \sum_{j \in N} x_i \Sigma_{i,j} x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in N} x_i = 1 \tag{2}$$

$$\forall i \in N \quad x_i \in [0, 1], \tag{3}$$

where $\boldsymbol{x} \in [0, 1]^N$ is the vector of assets allocation and $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ is the covariance matrix. Notice that objective function (1) is the portfolio variance $\sigma_p^2 = \boldsymbol{x}^T \boldsymbol{\Sigma} \boldsymbol{x}$ and Constraint (2) imposes that 100% of the portfolio is allocated. If short-selling is allowed, Constraint (3) can be discarded.

Indeed, the optimal solution of the GMVP problem can be derived analytically. However, additional constraints may be added to the basic model so our methodology is based on deriving a formulation that can be solved with a Mixed-Integer Linear Programming (MILP) solver like CPLEX [6]. For example, let $E[R_i]$ be the expected return of asset $i \in N$. It is possible to impose a target expected return μ_p for the portfolio in the GMVP problem by adding the following constraint:

$$\mu_p = \sum_{i \in N} R_i x_i. \tag{4}$$

This will identify a new portfolio in the efficient frontier. A graphical representation of the efficient frontier without risk-free asset, sometimes called *Markowitz's bullet*, and GMVP is depicted in Figure 1. Notice that the portfolios in the efficient frontier are those with an expected return larger than or equal to that of GMVP.

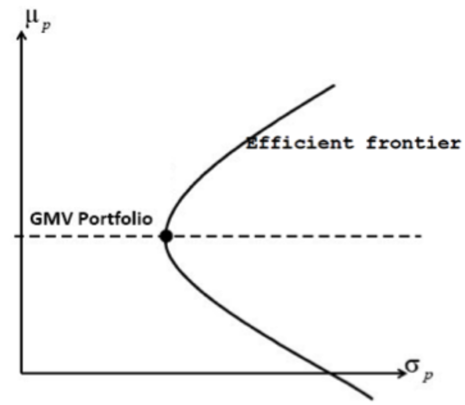

Figure 1: Efficient frontier (no risk-free asset) and global minimum variance portfolio.

Interestingly, the *Mutual Funds Theorem* states that any portfolio in the efficient frontier can be derived as a linear combination of any two distinct portfolios in the efficient frontier [9].

When a risk-free asset is available, i.e., an asset with expected return r_f and volatility equal to 0, thanks to the Mutual Fund Theorem [9] the efficient frontier is the line (called Capital Allocation Line - CAL) that can be computed by connecting the risk-free asset with the *tangency portfolio*, the latter being the tangent point in the Markowitz's bullet. This holds if r_f is smaller that the expected return of the GMV portfolio μ_{GMV}. A graphical representation of the CAL is provided in Figure 2. Let $E[R_t]$ the expected return of the tangency portfolio, and σ_t its standard deviation. The efficient frontier can be defined using the equation of the CAL:

$$\mu_p = r_f + \sigma_p \frac{E[R_t] - r_f}{\sigma_t}, \tag{5}$$

where the pair (σ_p, μ_p) identifies the portfolios in the efficient line.

It can be shown that the tangency portfolio is the one maximizing the Sharpe ratio, hence it can be found by solving the following nonlinear optimization prob-

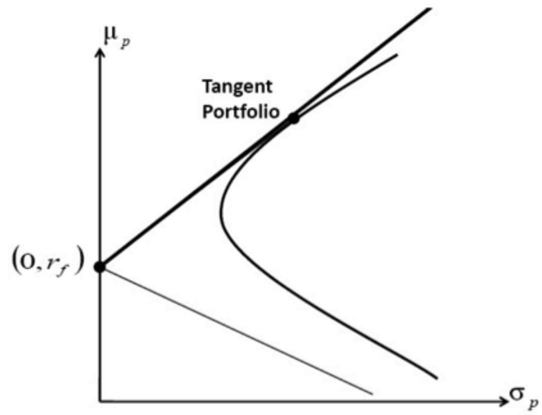

Figure 2: Efficient frontier with risk-free asset and tangency portfolio.

lem:

$$\max \quad \frac{\mu_t - r_f}{\sigma_t} \tag{6}$$

$$\text{s.t.} \quad \sum_{i \in N} x_i = 1 \tag{7}$$

$$\mu_t = \sum_{i \in N} R_i x_i \tag{8}$$

$$\sigma_t = \sqrt{\sum_{i \in N} \sum_{j \in N} x_i \Sigma_{i,j} x_j} \tag{9}$$

$$\forall i \in N \quad x_i \in [0, 1], \tag{10}$$

where the objective function (6) is the Sharpe ratio of the portfolio. Notice that this problem is nonlinear and nonconvex, so the MILP solver CPLEX cannot be used. Instead, global nonlinear solvers like SCIP [1], BARON [10, 11] or Couenne [2] should be used.

It is interesting to notice the relationship between the equation of the CAL (5) and the Capital Asset Pricing Model (CAPM) used to determine the expected return for an asset:

$$E[R_i] = r_f + \beta_i (E[R_m] - r_f), \tag{11}$$

where β_i is a measure of the reaction of the asset to a movement in the overall market, and $E[R_m]$ is the expected return of the market. It can be seen that in Equation (11) the market plays the role of the tangency portfolio in Equation (5). Hence, β_i can be interpreted as the standard deviation of asset i over the standard deviation of the market.

2.2 Black-Litterman model

Often a fund manager or investor has got some ideas concerning the future performance of the assets composing the portfolio. Those can be absolute values of future returns, or relative excess return of a set of assets compared to other assets. Unfortunately, it is not easy to express these view using the classical portfolio optimization methods presented in the previous section.

Black and Litterman [3, 4] proposed a way to overcome this issue by providing a formula that takes into account the investor views and can then produce an updated vector of expected returns. Starting from a vector of (prior) implied equilibrium returns $\mathbf{\Pi}$, the Black-Litterman formula derives the vector of (posterior) expected returns $E[\boldsymbol{R}]$ as follows:

$$E[\boldsymbol{R}] = \left[(\tau\mathbf{\Sigma})^{-1} + \boldsymbol{P}^T\mathbf{\Omega}^{-1}\boldsymbol{P}\right]^{-1}\left[(\tau\mathbf{\Sigma})^{-1}\mathbf{\Pi} + \boldsymbol{P}^T\mathbf{\Omega}^{-1}\boldsymbol{Q}\right], \tag{12}$$

where τ is a scalar, $\boldsymbol{P} \in \mathbb{R}^{K\times N}$ is a matrix that identifies the K absolute and/or relative views of the investor, $\mathbf{\Omega} \in \mathbb{R}^{K\times K}$ is a diagonal matrix expressing the uncertainties in each view, $\boldsymbol{Q} \in \mathbb{R}^{K}$ is the vectors of views, and $\mathbf{\Sigma} \in \mathbb{R}^{N\times N}$ is the covariance matrix of excess returns. A short description of these terms is presented in Section 2.2.1.

The Black-Litterman Formula (12) is basically a combination of the implied equilibrium returns and the investor views, where the latter can be defined partially and not for each asset combination. After the updated vector of returns $E[\boldsymbol{R}]$ has been computed, it can be used in some classical portfolio optimization framework to obtain an asset allocation more aligned with the investor views.

2.2.1 Elements of the Black-Litterman model

The elements needed to compute the updated vector of expected returns, according to the Black-Litterman Equation (12), are described in the following.

- **Π**: this represents the implied equilibrium return vector, and it can be computed using the CAPM formula (11) where the benchmark is the market-capitalization weighted portfolio.

- $\boldsymbol{Q} \in \mathbb{R}^K$: each element of this vector reflects one of the views of the investor on one or some assets, and these views can be absolute or relative. For example, if the first view is that asset i will have an absolute excess return of 4.5%, then $q_1 = 4.5$. At the same time, if the second view is that asset i will outperform asset j by 0.5%, then $q_2 = 0.5$. The link between these values and the corresponding assets is established by matrix P. Each view q_i is associated with a random error with mean 0 and variance ω_i. The latter value is included in the matrix Ω.

- $\boldsymbol{P} \in \mathbb{R}^{K \times N}$: each element $p_{i,j}$ of the matrix P identifies the assets involved in the investor views. More precisely, when the investor expresses the *absolute* view i on the asset j, then $p_{i,j} = 1$ and all the other elements in the row i of matrix P are set to 0. On the other hand, when the investor expresses a *relative* view i involving a set of assets J, then $p_{i,j}$, $\forall j \in J$ receive weights that are positive for outperforming assets and negative for underperforming assets, while the other elements of row j are set to 0 and the sum of the elements of row j is 0. In this case, the weights can be assigned using an equally weighting scheme (i.e., the weights assigned to the outperforming and underperforming assets are set to 1 over the number of outperforming assets and 1 over the number of underperforming assets, respectively) or they can be weighted using the market capitalization of the assets, as suggested in [7].

- $\boldsymbol{\Omega} \in \mathbb{R}^{K \times K}$: this is a diagonal matrix, where each element of the diagonal ω_i represents the variance of the error associated with the view q_i. If the confidence of view i is 100%, then ω_i should be 0. According to [7], the easiest way to calibrate the Black-Litterman model, which we follow in this work, is to define $\omega_i = (p_i \Sigma p_i^T)\tau$, where p_i is the $i-$th row of matrix P and Σ is the covariance matrix of excess returns. This way, the uncertainty on the view q_i is defined by the variance of the corresponding individual view portfolio $\omega_i = (p_i \Sigma p_i^T)$, and the value of the parameter τ becomes irrelevant, since it will change the entries of the matrix Ω but not the output of the Black-Litterman equation $E[\boldsymbol{R}]$.

2.3 Toy example

TO BE DONE - ADD SMALL EXAMPLE TO BETTER ILLUSTRATE THE ELEMENTS OF THE BL EQUATION

3 Implementation

We have implemented a Python library that provides the methods needed to derive an optimal portfolio using the Black-Litterman equation. This library exploit efficient data manipulation and processing packages like *numpy* and *pandas*, as well as the optimization library *Pyomo* [5], which allows define and solve (by exploiting state-of-the-art solvers) optimization problems in Python.

The functions implemented are the following:

- downloading data (e.g., adjusted close price, market cap.) given a list of tickers and a time frame
- computation of the global minimum variance portfolio given a list of assets
- computation of the tangency portfolio
- computation of the new vector of expected returns according to the Black-Litterman formula
- computation of the new optimal portfolio using the Black-Litterman expected returns.

- TO BE COMPLETED

4 Results

TO BE DONE

5 Conclusion

TO BE DONE

Acknowledgements

This research material has been prepared by NUS Invest. NUS Invest specifically prohibits the redistribution of this material in whole or in part without the written permission of NUS Invest. The research officer(s) primarily responsible for the content of this research material, in whole or in part, certifies that their views are accurately expressed and they will not receive direct or indirect compensation in exchange for expressing specific recommendations or views in this research material. Whilst we have taken all reasonable care to ensure that the information contained in this publication is not untrue or misleading at the time of publication, we cannot guarantee its accuracy or completeness, and you should not act on it without first independently verifying its contents. Any opinion or estimate contained in this report is subject to change without notice. We have not given any consideration to and we have not made any investigation of the investment objectives, financial situation or particular needs of the recipient or any class of persons, and accordingly, no warranty whatsoever is given and no liability whatsoever is accepted for any loss arising whether directly or indirectly as a result of the recipient or any class of persons acting on such information or opinion or estimate. You may wish to seek advice from a financial adviser regarding the suitability of the securities mentioned herein, taking into consideration your investment objectives, financial situation or particular needs, before making a commitment to invest in the securities. This report is published solely for information purposes, it does not constitute an advertisement and is not to be construed as a solicitation or an offer to buy or sell any securities or related financial instruments. No representation or warranty, either expressed or implied, is provided in relation to the accuracy, completeness or reliability of the information contained herein. The research material should not be regarded by recipients as a substitute for the exercise of their own judgement. Any opinions expressed in this research material are subject to change without notice.

References

[1] T. Achterberg. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009.

[3] F. Black and R. Litterman. Asset Allocation: Combining Investor Views with Market Equilibrium. Technical report, Goldman, Sachs & Co., 1990.

[4] F. Black and R. Litterman. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.

[5] William E. Hart, Jean-Paul Watson, and David L. Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

[6] IBM. *ILOG CPLEX 12.2 User's Manual*. IBM, 2010.

[7] Thomas Idzorek. *A step-by-step guide to the Black-Litterman model*, pages 17–38. Elsevier, 2007.

[8] H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

[9] R. C. Merton. An analytic derivation of the efficient portfolio frontier. *Journal of Financial and Quantitative Analysis*, 7(4):1851–1872, 1972.

[10] N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.

[11] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.