Fall 2018

Final Project Description and Requirements: Portfolio Optimization

Project Description

This problem is posted by McKinley Capital LLC as their Portfolio Optimization Horse Race for Spring/Summer 2019. The goal of the project is to construct a portfolio of U.S. stocks which tracks the expected return of a given benchmark index and yields the highest information ratio (as defined by the annualized monthly return divided by the standard deviation of the returns). The benchmark index is either the Standard & Poor's 500 index or the Russell 3000 Index from Jan. 2003 to Dec. 2017.

Courtesy to Dr. Guerard at McKinley, a set of more than 20 variables (a.k.a., factors) on a universe of all listed US and ADR stocks for 40 years are provided. These monthly factor data cover the period from 4/1976 to 12/2017. The data is contained in a zip file under Files on Canvas. The definitions of the factors are given in the Appendix and the README file.

Complete the following tasks:

1. Obtain the monthly returns of the Russell 3000 index from Jan. 2003 to Dec. 2017 and compute the information ratio and maximum drawdown of the buy-and-hold strategy for this index.

2. Consider the data in t = Dec. 2004, screen out the universe of investable securities by applying the following filters:
    a. Pick the top 4000 market-capitalization based stocks.
    b. Select stocks with ES values falling in the top 70-percentile among the 4000 stocks picked in a)

    Let R_i denote the monthly return of stock i and I_idx denote the monthly return of Russell 3000 index. Using the monthly return data over the past 24 months (i.e., a 24-month look-back period starting from and including Dec. 2004) in the data file and data collected in part 1, compute the expected returns of (R_i – I_idx) (denoted by μ_i), Var(R_i – I_idx), and Cov(R_i – I_idx, R_j – I_idx) (denoted by σ_{ij}) for all stock pairs (i, j) in the universe obtained in Dec. 2004.

3. Fit a 10-factor return prediction model as given by equation (15) in Reference [1] using all stocks contained in universe obtained in part 2. You may use any programming tools to fit this multivariate linear regression model across all securities, either Python or C++. Interpret the fitted results and explain whether the 11 fitted coefficients of the factors (a0, a1, a2, … , a10) obtained for Dec. 2004 are statistically significant.

4. Apply a Python package cvxopt to construct a Markowitz mean-variance optimal portfolio $w \equiv (w_1, w_2, \dots, w_N)$ where N is the number of stocks in the universe obtained in part 2. This portfolio maximizes the portfolio return based on the expected returns $\tilde{\mu}_i$ of all stocks predicted by the fitted model in part 3 (instead of the realized returns recorded in Dec. 2004). (Remark: to get robust predicted returns, it is suggested that one repeats the model fitting task described in part 3 for Nov. 2004, Oct. 2004, …, and Jan. 2004. Then use the average of the 12 sets of coefficients (a0, a1, a2, … , a10) to get the predicted returns in Dec. 2004.)

   The covariance matrix of returns (R_1 – I_idx, R_2 - I_idx, R3 – I_idx, … , R_N – I_idx), denoted by Σ, is constructed by setting the diagonal elements σ_{ii} to Var(R_i – I_idx), and off-diagonal elements σ_{ij} to Cov(R_i – I_idx, R_j – I_idx) for all stock pairs (for i != j) obtained in part 2 (remark: the covariance matrix Σ needs to be positive-definite. You shall write a module to check whether all the eigenvalues of Σ are positive. If Σ is not positive definite, then you may replace it with $\sqrt{\Sigma * \Sigma^T}$.). The constraint for this optimal portfolio is that the variance as computed by the above covariance matrix needs to be no greater than 0.0064 (which corresponds to a tracking error of no more than 8% with respect to the benchmark index). Namely, the portfolio weight vector w is the solution of the following mean-variance portfolio optimization problem:

$$max_w \sum_{i=1}^{N} w_i * \tilde{\mu}_i$$

$$\text{Such that:} \quad \textstyle\sum_{i=1}^{N} w_i = 1$$

$$w\Sigma w^T <= 0.0064$$

$$0 <= w_i <= U_i, \text{ for } i = 1, 2, \dots, N,$$

   where U_i is the upper limit of the weight of stock i to hold in the portfolio. Set $U_i = 0.1$ for all i.

   Reference [3] explains how to use the cvxopt package to solve for the above mean-variance optimal portfolio w. (Remark: an optional practical constraint is to require the weighted average of the 0-100 ranked ES value of the portfolio being no less than 80.)

5. Use the portfolio weight vector w obtained in Dec. 2004 to compute the portfolio return in Jan. 2005. Repeat part 2 – 4 for t = Jan. 2005, Feb. 2005, Mar. 2005, … , Nov. 2017, Dec. 2017 to obtain a series of 156 monthly returns of the monthly-optimized portfolio. Compute the information ratio and maximum drawdown of this portfolio strategy.

6. Consider a different portfolio strategy which maximizes the CTEF score of the portfolio subject to the same set of constraints as defined in part 4. This is simply done by repeating

part 4-5 for t = Jan. 2005, Feb. 2005, Mar. 2005, … , Nov. 2017, Dec. 2017 with the predicted returns of the stocks replaced by the CTEF scores observed in (t-1). Compute the information ratio and maximum drawdown of this portfolio strategy. Remark: this strategy is based on a simply return-prediction model which uses the CTEF score of a stock in month (t-1) to be its return in month t.

7. Implement a mean-variance optimal portfolio strategy based on your own return-prediction model which utilizes any machine learning/deep learning model to predict the stock returns in month t based on factor values observed in previous months. Namely, repeat part 3-5 with your own return-prediction model replacing the 10-factor model. Compute the information ratio and maximum drawdown of your portfolio strategy based on the 156 monthly returns.

   10 bonus points will be given if your strategy yields a higher information ratio and a lower maximum drawdown than the respective metrics of the strategies in part 5 and part 6.

Submission requirements:

1. Source codes shall be readily run, and tested for expected outputs without any issue. (These include all the C++/header files and Python scripts used in the project).

   The codes and implementation shall demonstrate proficiency in the following aspects:

   - Object-oriented design
   - Clean structure of the task flows and the corresponding functional modules
   - Adequate use of functions
   - Clear input/output interfaces
2. A written report describes the approaches taken, results found and analysis performed which pertain to each of the 7 parts in the project description.
   a. The construction of the input variables (i.e., features) and output variables to the price prediction models need to be clearly explained.
   b. The statistical validity and the accuracy of the price prediction models shall be provided.
   c. Clear justification of no look-forward bias in the models/strategies.
   d. Explicit description of the contribution of each group member in "Contribution" section. Under the name of each member, list the following information,
      i. the work done such as performed data cleaning/feature generation, built a Support Vector Machine regression model to predict price, conducted model validation with backtesting, etc.
      ii. the codes or parts of a code written by the member.

References

[1] Guerard, J.G., S.T. Rachev, and B.P. Shao (2013). "Efficient global portfolios: big data and investment universes", IBM J. RES. & DEV. Vol. 57, No. 5.

[2] Guerard Jr., J. B., Markowitz, H.M., & Xu, G. (2014). The role of effective corporate decisions in the creation of efficient portfolios, *IBM Journal of Research and Development* 58, No. 4, Paper 11.

[3] Blog: Markowitz Portfolio Optimization in Python. https://plot.ly/ipython-notebooks/markowitz-portfolio-optimization/

Appendix

Explanation of data labels in Total_Data5.csv

1.DATE

2.CUSIP

3.TICKER

4.PERMNO

5.GVKEY

6.MCAP - Market Capitalization

7.EP - Earnings/Price

8.BP - Book/Price

9.CP - Cash Flow/Price

10.SP - Sales/Price

11.DP - Dividend Yield

12.FEP1 - 1 year ahead IBES Forecasted EPS to Price/Last year's forecasted earnings per share

13.FEP2 - 2 year ahead IBES Forecasted EPS to Price/Last year's forecasted earnings per share

14.RV1 - FEP1 IBES Revisions

15.RV2 - FEP2 IBES Revisions

16.BR1 - IBES Breadth

17.BR2

18.CTEF - Consensus EPS I/B/E/S forecast, revisions and breadth

19.PM1 - price momentum as price(t-1)/price(t-12)

20.PM2 - price momentum as price(t-1)/price(t-7)

21.ES - (Eli Schwartz) Corporate Exports

22.RET - Returns

23.REP - Current EP/Average EP of last 5y

24.RBP - Current BP/Average BP of last 5y

25.RCP - Current CP/Average CP of last 5y

26.RSP - Current SP/Average SP of last 5y

27.RDP - Relative Dividend Yield

28.VOL - Monthly stock Volume

29.CRET - Monthly Stock Return

30.STATPERS - Date of IBES Forecast

31.USFIRM - US Firm (=1)

32.CURCODE - Currency Code

33.TOT - Total Number of FY1 Analysts

34.FGR1 - 1 year ahead forecast earnings per share monthly breadth

35.FGR2 - 2 year ahead forecast earnings per share monthly breadth

36.MRV1 - Mckinley definition of revisions in 2005

37.MRV2