MMF1921 – Operations Research

Project 2 (Summer 2020)

Due Date: 19–May–2020 by 9:00 AM

Please use MATLAB to solve this project. This is a group project, with groups of 2 or 3 students per group. Each group is expected to submit their own original work.

You are given:

- Raw market data consisting of adjusted closing prices for 20 U.S. stocks [1], and factor rates of return corresponding to 8 different factors [2].
- A MATLAB program template.
- Note: You are encouraged to reuse some of your code from Project 1 in this project.

You should hand in:

- A formal report composed of an introduction, methodology, analysis and conclusion.
- The MATLAB program and functions you wrote to solve this project.

⇒ Please submit all your files inside a compressed folder, including your report and MATLAB code. The name of the submitted file should be **FirstnameLastname.zip**. Only one submission per team is required. Please submit this file electronically through the Quercus portal (a printed copy is not required).

---

1 Introduction

The purpose of this project is to design an automated asset management system, i.e., an algorithmic trading system. In particular, this algorithm will be similar to the ones you prepared in Project 1, and should leverage some of the concepts, methods and optimization models seen throughout the course.

As part of the project, you are given some sample asset and factor data that you may use to train, validate and test your algorithm. This process should have you test multiple calibration and optimization models to develop your trading algorithm. In other words, you can develop your own investment strategy based on a model or ensemble of models that you believe will perform the best under different market environments (e.g., the U.S. market during the 2003 recession, global markets during the 2008 financial crisis, Asian markets during the recent bull market period, etc.). In addition, there are a few rules and limitations to narrow down the scope of this project. This is discussed in greater detail in Section 2.

This project has two main deliverables: a formal report detailing your model development, and a MATLAB program (and functions) containing your trading algorithm. As previously mentioned, your algorithm can be trained, validated and tested using the sample data provided. After submission, your algorithms will be tested on two different historical data sets to test their ability to adapt to different markets and market conditions. The algorithms will be assessed on their out-of-sample financial performance and their computational cost. From a financial perspective, they will be assessed by their out-of-sample Sharpe ratio and their average turnover rate. The computational cost will be assessed by runtime. This is discussed in greater detail in Section 3.

2 Investment competition rules

This project has a few rules in order to narrow down the scope of your project. These rules pertain to the input data frequency and type, the in-sample calibration and out-of-sample investment windows, and the required output. These rules will serve to guide your algorithm development for the subsequent competition. The rules are outlined below.

- **Input data**
  - The training data will consist of the same 20 stocks and 8 factors used for Project 1 (see Tables 1 and 2). However, a longer time period will be provided for you to train your models, ranging from Jan–2002 to Dec–2016. The factor data will include the risk-free rate.
  - The input data for the competition will be similar in format and type to the training data set. The data will consist of monthly asset prices and factor returns. Your MATLAB program should be able to accept different input data sets (i.e., the TA should be able to load different data sets in your program by changing the file names in the first section of the MATLAB template).
  - The competition time period will range anywhere from 1990 to 2019. During this period, two separate periods will be chosen from which to draw the competition data sets. Both periods will consist of a 10-year out-of-sample investment horizon (plus a 5-year period at the beginning for calibration)
  - The assets provided during the competition will consist of stocks and ETFs. The competition data sets will have anywhere between 15 to 40 assets.
  - The factors provided during the competition will be the same eight factors provided with the training data set, except their time period will align with that of the competition data set.
- **In-sample and out-of-sample windows**
  - Similar to Project 1, your algorithm cannot inspect data beyond the in-sample period when portfolio rebalancing takes place. In other words, your algorithm cannot check the future asset returns before investing (the algorithm must be 'blind' to the future). In addition, you are not allowed to embed any rules that discriminate based on knowledge of past events (e.g., you cannot 'hard code' your algorithm to be conservative during the period 2008–2009 based on your present-day knowledge that this is when the financial crisis took place).
  - Your first in-sample calibration window can be at most 5 years long. However, any subsequent calibration is allowed to use any new available data. For example, if the competition begins the out-of-sample investment horizon on Jan–2010, then the corresponding data set have data starting from Jan–2005. If your algorithm attempts to use data before this period, it will throw an error.
  - During the competition, the portfolios may be rebalanced either every 6 months (in January and July) or 12 months (in January). The rebalance frequency will be determined during the competition.
- **Required output: after being run, your algorithm should output**
  - A plot of your portfolio's cumulative wealth.
  - An area plot of your asset weights period-over-period.

- A table with the portfolio's out-of-sample average monthly excess return, monthly volatility, monthly Sharpe ratio calculated over the entire investment horizon, and average turnover rate. For example, if the investment horizon consists of a 10-year period, then you should provide a single value for the average monthly excess return, a single value for the observed volatility over the 10-year period, the ratio of excess return-to-volatility (Sharpe ratio), and the average turnover rate.

- Note 1: The turnover rate is calculated as the change in asset weights during each rebalance period. For example, if we rebalance our portfolio at time t, then the turnover is measured as the change in weights between the instant immediately before and after optimization takes place. For your convenience, the calculation of the portfolio turnover is already embedded into the MATLAB template.

- Note 2: Your MATLAB program should not produce more output than that requested. In other words, the program should not print hundreds of lines to the console for no reason. This is to make the desired output easy to find and read. The only exception will be warning messages intrinsic to MATLAB and/or Gurobi that you are unable to suppress.

Table 1: List of assets by ticker

F	CAT	DIS	MCD	KO	PEP	WMT	C	WFC	JPM
AAPL	IBM	PFE	JNJ	XOM	MRO	ED	T	VZ	NEM

Table 2: List of factors

Market ('Mkt_RF')	Size ('SMB')	Value ('HML')	Short-term reversal ('ST_Rev')
Profitability ('RMW')	Investment ('CMA')	Momentum ('Mom')	Long-term reversal ('LT_Rev')

3 Competition assessment criteria

The competition will test your trading algorithms using two previously unseen data sets. Thus, your submitted MATLAB code must be able to accept different data sets. For reproducibility, the new data sets will have the same format as the training data set provided with this project.

The assessment over these two data sets will be referred to as 'Trial 1' and 'Trial 2'. The trials will vary in number of assets, rebalance frequency (either 6 or 12 months), and investment period (anywhere between 1990 and 2019). However, the length of the investment horizon will be the same in both cases: a 10-year out-of-sample window. Thus, for example, if we have 6-month rebalancing, we will have 20 portfolio rebalancing periods. The assessment criteria is shown below in order of importance.

1. Ex-post Sharpe ratio over the entire investment horizon (higher is better).
2. Average turnover rate (lower is better).
3. Computational runtime (lower is better). The runtime will only become an issue if your trading algorithm takes more than (approximately) 20 minutes to run during each trial. For example, an algorithm that takes 30 minutes to run will be penalized. However, algorithms that run

sufficiently fast will not be penalized (e.g., from an evaluation perspective, there is no difference between an algorithm that runs in 5 minutes versus 8 minutes).

Your model development should incorporate at least some elements of the material seen in class (e.g., factor models, portfolio optimization), even if these elements do not make it into your final version of your trading algorithm (e.g., you may choose not to use factor models if they are not beneficial to your algorithm). You are welcome to incorporate methods exogenous to the course, such as machine learning methods, if they are beneficial to your algorithm. Remember that you must document and explain your methodology in your final report.

Finally, please note that the majority of your grade is not dependent on your algorithm's performance during the competition. The majority of your grade depends on your submitted report detailing your model development and testing, as well as the structure and readability of your code. Thus, you can expect to do well in the project if you make a good effort during your model development. On the other hand, simply having a 'winning' algorithm without justification through a proper model development report does not guarantee you will earn a good grade. For more information, please find the list of deliverables in the next section.

4 Deliverables

4.1 Report (70%)

Prepare a formal report, including an introduction, methodology, model selection process and testing, and a conclusion outlining the strengths and weaknesses of your algorithm and why you chose it. The report should demonstrate your understanding of finance and optimization theory, and reflect your knowledge of the material we have seen in class.

The report is worth 70% of the total. The distribution is the following

- Formal report structure and presentation: 10%
- Methodology: 20%
- Analysis from training, validation and testing: 25%
- Discussion and conclusion: 15%

4.2 MATLAB program (30%)

Prepare a MATLAB program and functions to perform the computational experiment. Use the main template provided. However, you are allowed modify this template as much as you see fit. In addition, you are allowed (and encouraged) to create your own functions if needed.

Be sure to properly comment on your code to briefly explain what you are doing. Your code should be easy to read and the TA should be able to run it. The TA will not debug your code and should not have to search for the results within the code.

The first section of tour 'MMF1921_Project_2_Main.m' program must be in the standard format provided to allow the TA to easily change the input historical data (for both assets and factors), time periods, and any other relevant parameter noted in the template provided. Please do not modify this section of the code.

The MATLAB program is worth 30% of the total, and it includes the assessment of the performance of your algorithm. The distribution is the following

- Properly structured and commented code: 10%
- Score from 'Trial 1': 10%
- Score from 'Trial 2': 10%

References

[1] Quandl.com. *Wiki – Various End-Of-Day Stock Prices*. https://www.quandl.com/databases/WIKIP/usage/export. [Online; accessed 07-Nov-2017]. 2017. (Visited on 11/07/2017).

[2] K. R. French. *Data Library*. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html. [Online; accessed 01-Feb-2020]. 2020. (Visited on 2020).