ISyE 6785 Final Project

Submission Description:

- Yifei Fan, ISyE 6785 Final Project.pdf: Report for ISyE 6785 final project.
- stats.xlsx: Results on annual average returns, standard deviations and Sharpe Ratios under different MA lag lengths and operation types.
- final project.pdf: presentation slides for this project
- paper.pdf: The reference paper: *A New Anomaly: The Cross-Sectional Profitability of Technical Analysis*.
- presentation.pdf: The presentation slides used for the presentation published on Tegrity.
- stockuniv.csv: A list of stock tickers that we can download historical price data from yahoo finance. These stocks make the universe of this project.
- stockvolatility.xlsx: The volatility of each stock in the stock universe. This is the intermediate result for constructing the volatility decile portfolios.
- Data (folder): The raw data used for this project are located in this folder. In order to make the python script work, please make sure that Data (folder), stockuniv.csv and moving_average.py are in the same directory.
  Files in Data (folder) include:
  - [stockname].csv': Historical price records for a stock whose name is given by [stockname].
  - nasdaq.csv: A list of stock tickers that belong to Nasdaq. (downloaded from internet)
- moving_average.py: Python script for constructing volatility decile portfolios and calculating average return and standard deviation of the portfolios.
- datacollector.py: Python script for collecting and storing historical price of stocks.

How to run the program?

1. Unzip 'Data.zip', put the 'Data' folder, stockuniv.csv and moving_average.py in the same directory.
2. Open 'moving_average.py', and set the lag length of the moving average at line 17.
3. Select the operation type by comment/uncomment the 'calPortRet' function from line 100 to line 132. There are two versions of calPortRet, one is for operating based on single stock, the other is for operating based on the entire portfolio. Please make sure that only one of them is working.
4. Run 'moving_average.py'
5. Repeat 2→4 to get the result for the different cases with different lag length and different operation types. If you're using spyder IDE, you can comment line 167 to line 171 after the first run because the tables are already in the memory pool.

Background Introduction

The project is an implementation of the trading algorithm basing on moving average timing strategy which was published in the paper *A New Anomaly: The Cross-Sectional Profitability of Technical*

Analysis. The paper was published by Yufeng Han, Ke Yang and Guofu Zhou in August, 2010. Prior to this work, technical analysis was widely used in the industry in that: (1) all major brokerage firms publish technical commentary on market and many of the advisory services are based on technical analysis; (2) Many top traders and investors use it to partially or exclusively (e.g., Schwager, 1993, 1995; Covel, 2005; Lo and Hasanhodzic, 2009). Recent studies, such as Brock, Lakonishok, and Lebaron (1992) and Lo, Mamaysky, and Wang (2000), find strong evidence of profitability of using technical anlaysis, primarily the moving averages to forecast the market. Zhu and Zhou (2009) further demonstrate that technical analysis can be a valuable learning tool about the uncertainty of market dynamics.

In this paper, the research group apply technical analysis to volatility decile portfolios. They view technical analysis as one of the signals investors use to make trading decisions. In particular, their technical analysis focus on applying the popular moving averages to time investments.

Data Source and Data Preprocessing

Historical prices of 1976 stocks from Nasdaq, starting from Jan 5th, 2010 to Dec. 31st, 2014 are used to test the consistency of moving average timing trading strategy established by the paper. The historical price data are downloaded from yahoo finance.

The data shall be preprocessed before calculation in the following three aspects: (1) a name list of available stocks is generated during downloading for future iteration; (2) stocks with incomplete data records are completed or deleted so that there are 1301 pieces of record for every stock; (3) the start index of each year and the number of business days of each year is calculated. All data preprocessing procedures are aimed for reducing the complexity of iterations on dates.

When implementing using python, we follow the following data handling philosophy: (1) read from file only once; (2) use sequential/ordered access as much as possible; (3) form the database at the beginning before making any logic judgments, calculating tables of historical daily returns, moving averages, volatilities and decile portfolio list.

Key Data Structure and Algorithm

The table of historical prices, daily returns and moving averages are stored in python dictionaries. In the dictionary, the keys are the stock names and the values are the historical values represented using a pandas.TimeSeries. When retrieving in the dictionary, the time complexity is $O(1)$.

Key	Type	Size	Value
AAL	TimeSeries	(1257,)	Date 2010-01-05 5.261844
AAME	TimeSeries	(1257,)	Date 2010-01-05 1.324163
AAON	TimeSeries	(1257,)	Date 2010-01-05 5.399662
AAPL	TimeSeries	(1257,)	Date 2010-01-05 28.767737
AAWW	TimeSeries	(1257,)	Date 2010-01-05 40.310001
AAXJ	TimeSeries	(1257,)	Date 2010-01-05 52.461069
ABAX	TimeSeries	(1257,)	Date 2010-01-05 24.023870
ABCB	TimeSeries	(1257,)	Date 2010-01-05 7.223898
ABCD	TimeSeries	(1257,)	Date 2010-01-05 4.01
ABCO	TimeSeries	(1257,)	Date 2010-01-05 15.830000

Volatility decile portfolios for a year are calculated at the end of the previous year. When calculating, first estimate the volatility of the stock using the daily returns before the new year; then sort the portfolios into 10 groups according to their volatilities. The results are stored in a table called ‘vlt_tb.

Index	Type	Size	Value
0	list	5	['VGSH', 0.000841...308872521, 0.000816...994074642, 0.000719...112598488, 0.000 ...
1	list	5	['VCSH', 0.001381...127657993, 0.001414902452611475, 0.001258...405838237, 0.001 ...
2	list	5	['VGIT', 0.003466...916487925, 0.003355...802449957, 0.002976...558016987, 0.002 ...
3	list	5	['VMBS', 0.005194...014035524, 0.004047...020805696, 0.003366...446418579, 0.003 ...
4	list	5	['VCIT', 0.003753...358775803, 0.003630...056400203, 0.003233...613961433, 0.003 ...

The result of decile

portfolios every year is recorded in a list of lists called ‘vltDecsPorts_ll’.

ıde ▲	Type	Size	Value
0	list	10	[['VGSH', 'VCSH', 'VGIT', 'VCIT', 'VMBS', 'CHSCP', 'ISHG', 'IGOV', 'VCLT', 'VGLT ...
1	list	10	[['VGSH', 'VCSH', 'VGIT', 'VCIT', 'VMBS', 'ISHG', 'IGOV', 'VCLT', 'CHSCP', 'PVTB ...
2	list	10	[['VGSH', 'VCSH', 'VGIT', 'VCIT', 'VMBS', 'ISHG', 'IGOV', 'VCLT', 'CHSCP', 'PVTB ...
3	list	10	[['VGSH', 'VCSH', 'VGIT', 'VMBS', 'VCIT', 'ISHG', 'IGOV', 'VCLT', 'PVTBP', 'VGLT ...

ıde ▲	Type	Size	Value
0	list	197	['VGSH', 'VCSH', 'VGIT', 'VCIT', 'VMBS',…
1	list	197	['JBHT', 'COKE', 'MNST', 'SP', 'ENDP', '…
2	list	197	['ACIW', 'SMBC', 'BCPC', 'RDI', 'NEOG', …
3	list	197	['HUBG', 'GUID', 'ABAX', 'PLCE', 'BWINB'…
4	list	197	['SHEN', 'GABC', 'ISIG', 'SILC', 'ASMI',…
5	list	197	['REXI', 'PBCP', 'RGEN', 'AIMC', 'ENG', …
6	list	197	['CRZO', 'CTG', 'LONG', 'MLNX', 'SGI', '…
7	list	197	['NSEC', 'NEO', 'HDNG', 'SBGI', 'NFLX', …
8	list	197	['TSYS', 'SNTA', 'TENX', 'MNKD', 'LIOX',…
9	list	203	['VRML', 'LRAD', 'TUES', 'THLD', 'WGBS',…

ıde ▲	Type	Size	Value
0	str	1	VGSH
1	str	1	VCSH
2	str	1	VGIT
3	str	1	VCIT
4	str	1	VMBS
5	str	1	CHSCP
6	str	1	ISHG

The mathematical form of daily return on portfolio j, on day t, with the lag length of moving average L is the following.

$$\tilde{R}_{jt,L} = \begin{cases} R_{jt}, & if \ P_{jt-1} > A_{jt-1,L} \\ r_{ft}, & otherwise \end{cases}$$

where R_{jt} is the daily return on day t for portfolio j, P_{jt-1} is the adjusted closing price for portfolio j on day $(t-1)$, $A_{jt-1,L}$ is the moving average of portfolio j’s adjusted closing price with lag length L, and r_{ft} is the risk free rate (0.01% was used).

According to the trading algorithm, we define the key function ‘calPortRet()’ which can calculate the return of a portfolio on a specific year. The input of this function include: a name list of stocks in that portfolio, a year number specifying which year it is, and the database of historical prices, daily returns and moving averages we calculated at the beginning. The output of this function is the average daily return of the portfolio for that year. There are two versions of this functions in the python script, one is for the case when operations are based on single stock, the other is for when operations are based on whole portfolio. (To get the price, daily return and moving average for a portfolio, we simply take the unweighted average among the members of that portfolio.)

Results

After execution, you can view the result in spyder’s variable explorer. Here is a table of the contents of the variables. You can view the statistics (average, standard deviation and Sharpe ratios) in ‘stats.xlsx’.

Table 1 Variable list

Name	Contents
price_db	A dictionary of historical adjusted price of all stocks
dreturn_db	A dictionary of historical daily return of all stocks
mvavr_db	A dictionary of historical moving average of all stocks
vlt_tb	A table of stock volatility estimated at the end the each year
vltDecsPorts_ll	The construction of volatility decile port of each year
MADecsRets	The daily returns of volatility decile portfolio every year during the test period
MADecsVars	The standard deviation of volatility decile portfolio daily return every year
Rets	Average annual returns for volatility decile portfolios
Vars	Standard deviation on annual returns for volatility decile portfolios

Here are the graphs we generated from the result.

Here are the average annual returns under different MA lag length conditions when operation is based on portfolios.

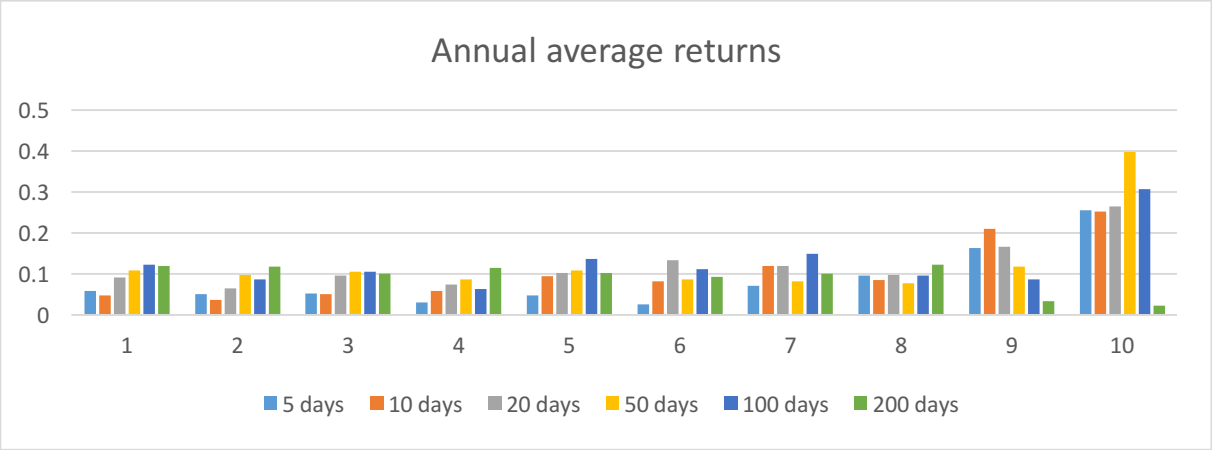

Figure 1 Average annual returns for volatility decile portfolio with different MA lag length, operation based on portfolios

Here are the average annual returns under different MA lag length conditions when operation is based on single stocks.

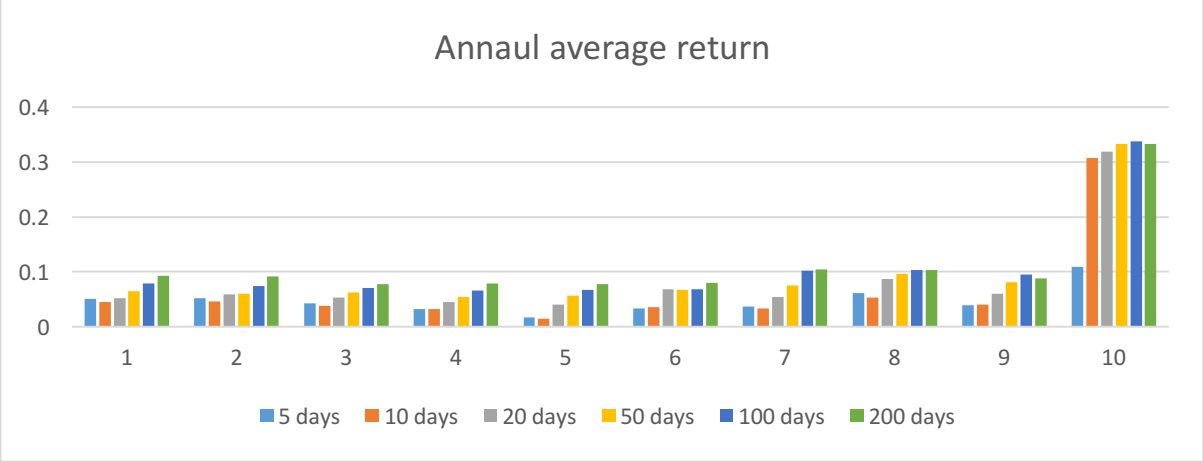

Figure 2 Standard deviation on annual returns for volatility decile portfolio with different MA lag length, operation based on single stocks

Here are Sharpe ratios under different MA lag length conditions when operation is based on portfolios.

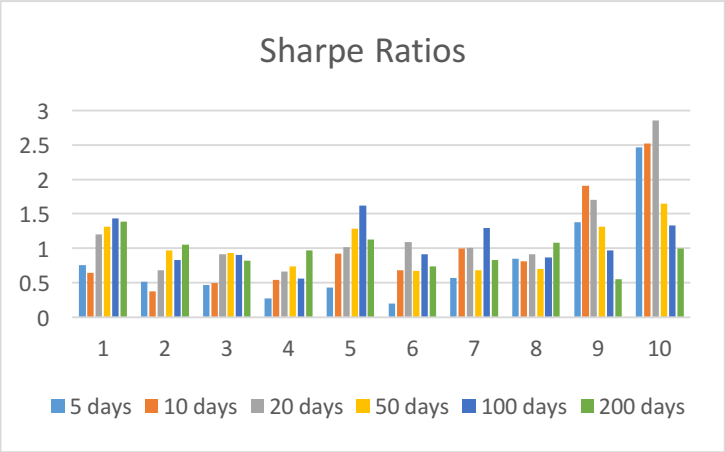

Figure 3 Sharpe ratios under different lag length, operation based on portfolios

Here are Sharpe ratios under different MA lag length conditions when operation is based on single stocks.

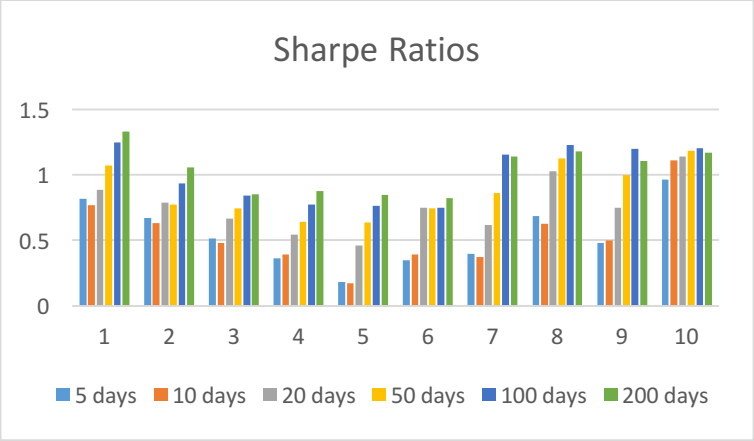

Figure 4 Sharpe ratios under different MA lag length conditions, operations based on single stocks

Observations

Roughly speaking, decile portfolios with high volatility can take more advantage on moving average timing strategy, especially for the highest volatility decile portfolio. This is true in regardless of whether operations are based on portfolios or on single stocks. However, as the moving average length gets larger, the differences between the returns of decile portfolios get smaller.
When operation is based on single stock, average return is slightly higher for longer MA period.

Conclusion

The performance of test period is not as good as training period, when average return grows almost linearly with volatility. Nevertheless, certain characteristics still holds. Generally speaking, the highest volatility decile portfolio still gets the highest return under same conditions. Therefore, when applying the moving average timing strategy, it is better to utilize stocks with high volatilities because its return and Sharpe ratio are higher.