Technical Guidance

March 22, 2020

This document is the technical guidance for the group project of Dr. Xin Tong's DSO 530 class in Spring 2020 at the University of Southern California. It provides some details about the common features that we created for each group. It will also give you some details about the metrics to evaluate your trading strategy.

You are provided with 2 zip files. The first zip file (raw data.zip) contains the daily stock data of the 50 companies contributing to the SSE Index. The second zip file (engineered features.zip) contains the features engineered from the raw data and the feature engineering process is described below. You are encouraged to engineer more predictors from raw data to achieve better performance.

Note that all data features in the second zip file (engineered features.zip) are lagged by one day because you are not allowed to use today's data to predict today's value.

1 Common Features

From the raw datasets, we have engineered the following variables: *MA5* (*5-day Moving Average*), *MA15* (*15-day Moving Average*), *5-day Return*, *15-day Return*, *RSI* (*Relative Strength Index*), *Stochastic %K* (*Stochastic Oscillator*), *Stochastic %D* (*Stochastic Oscillator*), *Chaikin AD Line*, *PROC* (*Price Rate of Change*), *OBV* (*On Balance Volume*). These are standard features used in technical analysis of the stock market and all of them can be constructed by daily data.

The detailed definitions of all features are as follows.

Note that all the C_t below represent the *adjusted closing price* at time t.

A **moving average** (**MA**) is widely used in technical analysis. It helps smoothing out price action by filtering out the "noise" from random short-term price fluctuations. It is a trend-following indicator. In particular, in the project, we provide two features based on the MA idea:

$$\text{MA5(t)} = \frac{1}{5}\sum_{i=0}^{4} C_{t-i},$$

$$\text{MA15(t)} = \frac{1}{5}\sum_{i=0}^{14} C_{t-i}.$$

N-Day Returns

$$\text{5-Day Return(t)} = \frac{C_t - C_{t-4}}{C_{t-4}},$$

$$\text{15-Day Return(t)} = \frac{C_t - C_{t-14}}{C_{t-14}}.$$

The **relative strength index** (**RSI**) is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset. The RSI is displayed as an oscillator (a line graph that moves between two extremes) and can have a reading from 0 to 100. The indicator was originally developed by J. Welles Wilder Jr. and introduced in his seminal 1978 book, *New Concepts in Technical Trading Systems*.

$$\text{RSI(t)} = 100 - \frac{100}{1 + \text{RS}(t)},$$

where

$$\text{RS}(t) = \frac{\text{average gain(in percentage) over the 14 days leading up to and including t}}{\text{average loss(in percentage) over the 14 days leading up to and including t}}.$$

A **stochastic oscillator** is a momentum indicator that compares a particular closing price of a security to a range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result. It is used to generate overbought and oversold trading signals, utilizing a 0-100 bounded range of values. We consider two stochastic oscillators:

$$(1) \qquad \%K(t) = \frac{C_t - L_{t,14}}{H_{t,14} - L_{t,14}},$$

where $L_{t,14}$ and $H_{t,14}$ are the highest high and lowest low during the 14 days leading up to and including t.

$$(2) \qquad \%D(t) = \text{3-period moving average of \%K}.$$

The **accumulation/distribution line** (**Chaikin AD Line**) was created by Marc Chaikin to determine the flow of money into or out of a security.

$$\text{AD}(t) = \text{CLV}(t) \times \text{Volume}(t),$$

where

$$\text{CLV}(t) = \frac{(C_t - L_t) - (H_t - C_t)}{H_t - L_t}.$$

Note that **close location value** (**CLV**) looks at the location of the close and compares it to the range for a given period (one day, week or month). Here, our definition specifically uses one day.

Price Rate of Change (**PROC**): equivalent to n-day return and we use n=1 here.

On-balance volume (**OBV**) is a technical trading momentum indicator that uses volume flow to predict changes in stock price. Joseph Granville first developed the OBV metric in the 1963 book *Granville's New Key to Stock Market Profits*.

$$\mathrm{OBV}(t) = \mathrm{OBV}(t-1) + \begin{cases} \mathrm{Volume(t)}\,, & \text{if } C_t > C_{t-1} \\ 0\,, & \text{if } C_t = C_{t-1} \\ -\mathrm{Volume(t)}\,, & \text{if } C_t < C_{t-1} \end{cases}. \tag{1}$$

These features of each stock have been created for you. As an illustration, we calculate the *N-day Returns* of the one stock as follows.

[1]:
```
import pandas as pd
import numpy as np

# read csv file
Stock = pd.read_csv('SH600000.csv')
Stock
```

[1]:

	Time	Open	High	Low	Close	Volume
0	Day1	154.6414	159.7405	154.4560	156.9592	45036400
1	Day2	156.9592	158.7207	156.3102	156.5883	21043100
2	Day3	156.5883	158.5353	155.8467	158.4426	23335200
3	Day4	158.9988	162.1510	158.9988	159.5551	33835300
4	Day5	159.5551	160.8530	158.6280	160.0186	29530100
..	…	…	…	…	…	…
751	Day752	157.8890	159.6702	157.5073	158.0162	40150148
752	Day753	158.5252	158.5252	154.8356	155.2172	37033892
753	Day754	155.3445	156.3623	155.2172	156.2351	21671028
754	Day755	156.3623	156.3623	155.2172	155.7262	13678175
755	Day756	155.9806	156.7440	155.5989	156.3623	15739054

[756 rows x 6 columns]

[2]:
```
price = Stock["Close"]
# initialize results array
day5Return = np.zeros(len(Stock))
day15Return = np.zeros(len(Stock))

# use formular to calculate the results
for i in range(14,len(Stock)):
    day5Return[i] = 100*(price[i]-price[i-4])/price[i-4]
    day15Return[i] = 100*(price[i]-price[i-14])/price[i-14]
```

Because we are not allowed to use today's data to predict today's value, we move the whole data set one day forward.

[3]:
```
#calculate the lagged data
day5Returnlag = np.append(np.array([0]),day5Return[0:len(day5Return)-1])
day15Returnlag = np.append(np.array([0]),day15Return[0:len(day5Return)-1])
```

[4]:
```
Stock["day5Returnlag"] = day5Returnlag
Stock["day15Returnlag"] = day15Returnlag
```

[5]:
```
Stock.loc[16:30,["Time","day5Returnlag"]]
```

[5]:

	Time	day5Returnlag
16	Day17	0.233345
17	Day18	-3.690899
18	Day19	-4.745577
19	Day20	-3.785677
20	Day21	-5.238613
21	Day22	-2.275446
22	Day23	-2.821127
23	Day24	-1.997545
24	Day25	-0.184254
25	Day26	-1.102947
26	Day27	-0.617661
27	Day28	-0.741179
28	Day29	-0.246192
29	Day30	0.991346
30	Day31	1.491572

For all engineered variables, we only give the values stating at Day17.

2 Evaluation Metrics

As decribed in project description, we suggest 3 metrics to evaluate your trading strategy.

The first one is the amount of money you have at the end of Day756.

The second metric is the *Sharpe ratio*. Usually, people prefer a higher Sharpe ratio.

The third metric is the number of days you make a profit. This metric reflects some psychological effects of your trading algorithm.

The first and third metrics are straightforward to calculate. Here, we talk more about the *Sharp ratio*.

The *Sharpe ratio* was developed by Nobel laureate William F. Sharpe and is used to help investors understand the return of an investment strategy compared to its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility.

Here, to simplify the problem, we set the risk free rate to zero and use the following formula to calculate the Sharp Ratio:

$$Sharp\ ratio = \frac{\sqrt{252} \times Mean(pnl)}{StdDev(pnl)}$$

where

$$pnl(t) \ = \ \frac{total\ asset(t)}{total\ asset(t-1)} - 1$$

and

$$total\ asset(t) = money\ in\ investment\ account(t) + holding\ stock\ value(t)$$

Note that *pnl* represents *profit and loss* and *(t)* represents that it is the value at the day t.

3 A Toy Example

Suppose stock A's adjusted closing price (per share) within one week is as follows.

	Day1	Day2	Day3	Day4	Day5	Day6	Day7
Stock A	27	27.5	28	27	26	26.5	27

Assume that we have $10,000 on the evening before Day1. Suppose our investment strategy is to only trade Stock A and to execute the following operations:

(1) Buy 300 shares of Stock A at Day 1

(2) Sell 300 shares of Stock A at Day 3

(3) Buy 350 shares of Stock A at Day 5

(4) Sell 350 shares of Stock A at Day 7

Then we calculate our totol asset day by day:

Day 1: total asset(1) = (10000 - 300 * 27 * (1 + 0.065%)) + (300 * 27) = 9994.735

Day 2: total asset(2) = (10000 - 300 * 27 * (1 + 0.065%)) + (300 * 27.5) = 10144.735

Day 3: total asset(3) = (10000 - 300 * 27 * (1 + 0.065%)) + (300 * 28 * (1 - 0.065%)) = 10289.275

Day 4: total asset(4) = total asset(3) = 10289.275

Day 5: total asset(5) = (10289.275 - 350 * 26 * (1 + 0.065%)) + (350 * 26) = 10283.36

Day 6: total asset(6) = (10289.275 - 350 * 26 * (1 + 0.065%)) + (350 * 26.5) = 10458.36

Day 7: total asset(7) = (10289.275 - 350 * 26 * (1 + 0.065%)) + (350 * 27 * (1 - 0.065%)) = 10627.2175

Metric one is the total asset of the last day after selling all the holding stocks: $10627.2175

Metric two is the *Sharp ratio*:

We calculate the *pnl(t)* from the second day to the last day.

According to the formula above, we can get the results as follows.

pnl(2) = total asset(2) / total asset(1) - 1 = 10144.735 / 9994.735 - 1 = 0.015007901660224032

pnl(3) = total asset(3) / total asset(2) - 1 = 10289.275 / 10144.735 - 1 = 0.014247784688313558

pnl(4) = total asset(4) / total asset(3) - 1 = 10289.275 / 10289.275 - 1 = 0

pnl(5) = total asset(5) / total asset(4) - 1 = 10283.36 / 10289.275 - 1 = -0.0005748704354776 59

pnl(6) = total asset(6) / total asset(5) - 1 = 10458.36 / 10283.36 - 1 = 0.01701778407057608

pnl(7) = total asset(7) / total asset(6) - 1 = 10627.2175 / 10458.36 - 1 = 0.016145695883484 69

[6]:
```
pnl = np.zeros(6)
pnl[0] = 10144.735 / 9994.735 - 1
pnl[1] = 10289.275 / 10144.735 - 1
pnl[2] = 10289.275 / 10289.275 - 1
pnl[3] = 10283.36 / 10289.275 - 1
pnl[4] = 10458.36 / 10283.36 - 1
pnl[5] = 10627.2175 / 10458.36 - 1
```

[7]:
```
pnl
```

[7]:
```
array([ 0.0150079 ,  0.01424778,  0.        , -0.00057487,  0.01701778,
        0.0161457 ])
```

Sharp ratio = sqrt(252) * mean(pnl) / std(pnl) = 21.691564622961124

[8]:
```
Sharp_ratio = np.sqrt(252) * np.mean(pnl) / np.std(pnl)
Sharp_ratio
```

[8]:
```
21.691564622961124
```

Metric three is the number of days you make a profit (compared to yesterday): 4

[9]:
```
np.sum(pnl > 0)
```

[9]:
```
4
```

references:

https://www.investopedia.com/