

Paper 163-2012

Comparing Stock Returns Forecasting Methods Using SAS®

Wei Wang, University of Arizona, Tucson, AZ

Arthur Li, City of Hope National Cancer Center, Duarte, CA

ABSTRACT

The accuracy of forecasting stock returns is the key component to generating profits on Wall Street. There are many methods to forecasting stock returns. Almost all the data mining methods are concerned with creating analytical models that are based on historical data trends. Choosing the best forecasting method is essential to obtaining fruitful stock returns. SAS® provides a flexible platform that allows one to easily compare different forecasting methods. In this paper, we compare three most-commonly-used technical trading methods (Moving Average, Relative Strength Index, and Bollinger Bands) by using SAS based on historical stocks from 1980 to 2010. All the calculations are performed by utilizing the DATA step or SAS macros. The comparison generated from the program helps us to decide the best forecasting method.

INTRODUCTION

Common ways to perform stock analysis in the real world are fundamental analysis and technical trading analysis. This paper focuses on testing the efficiency of technical trading methods. Between the 1890s and the 1980s, scholars have proven that these methods are quite efficient, but not so much for the data after the 1980s. In this paper, we chose three of the most popular technical trading methods, Moving Average, Relative Strength Index, and Bollinger Bands, to compare their efficiencies based on data between Jan. 2, 1981 to Dec. 31, 2010 from the NYSE.

DATA SAMPLE

In order to maintain integrity and consistency, we will keep only those stocks that were actively traded during the entire 30 year period. Thus, 355 individual stocks “made the grade.” There is one more reason that we only tracked these 355 stocks: the size of our daily data sample is above 4G. Without the filter, the size of more than 1000 NYSE daily stock data would be too large for us to calculate on our computer.

When we choose a buy-and-hold strategy for these stocks, the average daily return of these 355 stocks is the benchmark of our technical analysis. Here are the first 5 observations of our data set. In this dataset, the DATE is the trading date, CUSIP is the stock ID, RET is the daily return of stock, and ADJPRC is the adjusted price for stock.

Obs	DATE	CUSIP	RET	adjprc
1	19810102	00036110	0.009174	4.07407
2	19810105	00036110	0.045455	4.25926
3	19810106	00036110	0.008696	4.29630
4	19810107	00036110	-0.103448	3.85185
5	19810108	00036110	-0.057692	3.62963

The average daily return of these 355 stocks is 0.0638% (with SD = 2.2624%). If the predicted return from any of the technical methods is greater than this threshold, it means that this method is proven to be efficient.

MOVING AVERAGE METHOD

Moving Average is one of the most popular and classical technical analysis methods. The basic idea of the moving average strategy is that a moving average line has a resistance or support effect. Also the investors can use this strategy very easily. For example, the 10-day moving average number for day n is the average closing price from day $n-9$ to day n . Usually, the 1-day, 2-day and 5-day moving average lines appear to be short period ones; the 50-day, 150-day and 200-day moving average lines appear to be long period ones. The essence of this strategy is that when the short moving average line cuts the long moving average line from below, it is a buy signal; otherwise, it's a sell signal. We assume the investor will hold or sell the stock until the next buy or sell signal exists; we call this

strategy a Variable-Length Moving Average (VMA). If an investor will hold or sell a stock after a fixed period, we call it a Fixed-Length Moving Average (FMA).

We define moving average by using the following program:

Program 1A:

```
data MA;
  set summer.sample;
  MA2 = adjprc;
  MA5 = adjprc;
  MA50 = adjprc;
  MA150 = adjprc;
  MA200 = adjprc;
run;

proc sort data = MA;
  by cusip date;
run;

%macro calave(days);
  %do i = 1 %to &days-1;
    data MA;
      set MA;
      MA&days = MA&days + lag&i(adjprc);
      if &i = &days-1 then MA&days = MA&days/&days;
      if &i = &days-1 then do;
        if &days = 2 and date<mdy(01,05,81) then MA2 = .;
        if &days = 5 and date<mdy(01,08,81) then MA5 = .;
        if &days = 50 and date<mdy(03,13,81) then MA50 = .;
        if &days = 150 and date<mdy(08,05,81) then MA150 = .;
        if &days = 200 and date<mdy(10,15,81) then MA200 = .;
      end;
    run;
  %end;
%mend;

%calave(2)
%calave(5)
%calave(50)
%calave(150)
%calave(200)
data summer.MovingAverage;
  set MA;
run;
```

The next step is to use the following code to test our VMA moving average data set. Here is an example where we test the 1-day/200-day, 1-day/150-day, and 1-day/50-day strategy.

Program 1B:

```
data MAresult;
  set summer.movingaverage;
run;

proc means data = MAresult noprint;
  var ret;
  output out = buyandhold n = BHn mean = BHmean std = BHstd;
run;

data MAresult;
  set MAresult;
  if adjprc=. or MA200=. then status = 0;
```

```

else if adjprc>MA200*1.01 then status = 1;      *VMA buy(1,200,0.01);
else if adjprc<MA200*0.99 then status = -1;    *VMA sell(1,200,0.01);
else status = 0;

if adjprc=. or MA50=. then status = 0;
else if adjprc>MA50*1.01 then status = 1;      *VMA buy(1,50,0.01);
else if adjprc<MA50*0.99 then status = -1;    *VMA sell(1,50,0.01);
else status = 0;

if adjprc=. or MA150=. then status = 0;
else if adjprc>MA150*1.01 then status = 1;    *VMA buy(1,150,0.01);
else if adjprc<MA150*0.99 then status = -1;  *VMA sell(1,150,0.01);
else status = 0;
run;

proc sort data = MAresult;
  by status;
run;

proc means data = MAresult noprint;
  var ret;
  by status;
  output out = buyandsell n = BSn mean = BSmean std = BSstd;
run;

data VMAfinal;                                *VMA final result!;
  merge buyandhold buyandsell;
  by _TYPE_;
  drop _FREQ_ _TYPE_;
  if status ~= 0 then
    t = (BSmean-BHmean)/sqrt(BHstd*BHstd/BHn+BHstd*BHstd/BSn);
run;

```

SAS Output (1-day/200-day):

BHn	BHmean ¹	BHstd	status ²	BSn ³	BSmean ⁴	BSstd	t
2684107	0.000638	0.022624	-1	941759	-0.001572	0.028421	-81.5864
2684107	0.000638	0.022624	0	242051	0.000192	0.018246	.
2684107	0.000638	0.022624	1	1500297	0.002098	0.018701	63.2915

1. The daily return of benchmark
2. -1 means sell signal, 1 means buy signal
3. Bsn is the number of how many buy or sell signals in this period
4. BSmean is the average return for buy or sell signal

SAS Output (1-day/150-day):

BHn	BHmean	BHstd	status	BSn	BSmean	BSstd	t
2684107	0.000638	0.022624	-1	963737	-0.001934	0.028006	-95.7500
2684107	0.000638	0.022624	0	255045	0.000288	0.017513	.
2684107	0.000638	0.022624	1	1465325	0.002391	0.018997	5.4302

SAS Output (1-day/50-day):

BHn	BHmean	BHstd	status	BSn	BSmean	BSstd	t
2684107	0.000638	0.022624	-1	989249	-0.003835	0.025150	-168.095
2684107	0.000638	0.022624	0	379454	0.000184	0.016024	.
2684107	0.000638	0.022624	1	1315404	0.004133	0.021618	45.144

In the above output, we can see a 1-day and 200-day moving average strategy can earn 0.2098% at buy side and -0.1572% at sell side, which is much higher than our benchmark of 0.0638%, which has almost the same result as the 1-day and 150-day moving average strategy and the 1-day and 50-day moving average strategy. Clearly, on average, investors can obtain higher profits when they choose a variable-length moving average strategy, whether they long sell a stock in a "buy" period or short sell

one in a “sell” period. We have essentially the same result for the Fixed-Length Moving Average (FMA).

RELATIVE STRENGTH INDEX (RSI) STRATEGY

RSI is another popular technical analytical strategy. The basic idea of the RSI strategy is very different from the moving average strategy. RSI strategy focuses on the total gain or loss in prior market days. From the total gain or loss information, investors will know whether the overbought or oversold phenomenon has existed. The detailed formula of calculating RSI is as follows:

- $RS = \text{Ave. Gain} / \text{Ave. Loss}$ in prior n days;
- $RSI = 100 - 100 / (1 + RS)$

From the formula above, we know that RSI is a number between 0 and 100. Usually, if $RSI > 80$, the investor “overbought” for several days and they should sell the stock; if $RSI < 20$, it the investor “oversold” and they should buy the stock.

In our paper, we use a 14-day RSI, which means we calculate the average gain or loss in the previous 14 days of trading.

This is how we define RSI in SAS:

Program 2A:

```
data RSI;
  set RSI;
  by cusip;
  ladjprc = lag(adjprc);
  if adjprc>ladjprc then gain = adjprc-ladjprc;
  if adjprc<ladjprc then loss = ladjprc-adjprc;
  if first.cusip then do;
    gain = .;
    loss = .;
  end;

  retain days 0;
  days+1;
  if first.cusip then days = 1;
  if gain = . then gain = 0;
  if loss = . then loss = 0;

  retain sumgain 0;
  sumgain = sumgain + gain;
  retain sumloss 0;
  sumloss = sumloss + loss;
  if days=15 then do;
    AveGain = (sumgain-gain) / 14;
    AveLoss = (sumloss-loss) / 14;
    RS = AveGain / AveLoss;
    RSI = 100-100/(1+RS);
  end;
  drop sumgain sumloss;
run;

%macro calRSI();
  %do i=16 %to 7570;
    data RSI;
      set RSI;
      lAveGain = lag1(AveGain);
      lAveLoss = lag1(AveLoss);
      if days = &i then do;
        AveGain = (lAveGain*13+gain)/14;
```

```

                AveLoss = (lAveLoss*13+loss)/14;
                RS = AveGain / AveLoss;
                RSI = 100-100/(1+RS);
            end;
        %end;
    %mend;

%calRSI;

data summer.RSIresult;
    set RSI;
run;

```

The following program is a sample for the Variable-Length RSI (VRSI). We define Variable-Length RSI strategy based upon an overbought signal (RSI>80); the investor short sells the stock until RSI<60; when an oversold signal (RSI<20) appears, the investor long sells the stock until RSI>40.

Program 2B:

```

data VRSIB;
    set RSIresult;
    by cusip;
    drop status;
    retain period 0;
    if first.cusip then period = 0;
    if RSI<20 then period = 1;          *RSI<20 buy and until RIS>40;
    if RSI>40 then period = 0;
    RealPeriod = lag(period);
    if RealPeriod = 0 then delete;
run;

data VRSIS;
    set RSIresult;
    by cusip;
    drop status;
    retain period 0;
    if first.cusip then period = 0;
    if RSI>80 then period = 1;          *RSI>80 sell and until RIS<60;
    if RSI<60 then period = 0;
    RealPeriod = lag(period);
    if RealPeriod = 0 then delete;
run;

proc means data = VRSIB;
    var ret;
run;

proc means data = VRSIS;
    var ret;
run;

```

SAS Output:

The MEANS Procedure					
Analysis Variable : RET Returns					
	N	Mean	Std Dev	Minimum	Maximum
	41247	0.0022246	0.0379587	-0.5000000	0.8773586

In the above output, we can see that the average daily return is 0.222246%, which is much higher than the benchmark. We also obtain a similar output utilizing a fixed-length RSI strategy. In other words, the RSI strategy was efficient during the last 30-year period.

BOLLINGER BANDS STRATEGY (BOLL)

Bollinger Bands (BOLL) is a technical analytical tool which was invented by John Bollinger in the 1980s. The basic idea of the Bollinger Bands Strategy is to find the range (band) of fluctuation of a stock's price. The price that is out of the range is considered abnormal; the price will eventually migrate back into this range. Bollinger Bands consist of:

- a middle band, consisting of an N-period simple moving average (MA)
- an upper band, consisting of K times an N-period standard deviation is above the middle band ($MA + K\sigma$)
- a lower band, consisting of K times an N-period standard deviation is below the middle band ($MA - K\sigma$)

Typically, the moving average uses a 20-day period ($N=20$), and the upper and lower bands are defined by using a double standard deviation ($K=2$), which is BOLL (20, 2). When the price of the stock is above the upper band, the sell signal appears; when the price is below the lower band, the buy signal appears. In this paper, we also try $K=4$, which represents 4 times the standard deviation.

We define the variable-length Bollinger Bands Strategy as when a buy/sell signal appears, the investor long/short sells the stock until the price of the stock is above/below the $MA-\sigma$ ($MA+\sigma$).

The following program illustrates how we define the Bollinger Bands Strategy in SAS:

Program 3A:

```
data BB;
    set summer.sample;
    MA20 = 0;
run;

proc sort data = BB;
    by cusip date;
run;

%macro calave(days);
    %do i = 1 %to &days;
        data BB;
            set BB;
            MA&days = MA&days + lag&i(adjprc);
            if &i = &days then MA&days = MA&days/&days;
            if &i = &days then do;
                if &days = 20 and date<mdy(02,05,81) then MA20 = .;
            end;
        run;
    %end;
%mend;

%calave(20)

%macro sd;
    proc delete data=all;
    run;

    %do firstobs = 1 %to (2684248-20);
        %let lastobs = %eval( &firstobs + 19);
        proc means data=BB(firstobs=&firstobs obs=&lastobs) noprint;
            var adjprc;
            output out=tempfile stddev=sd20;
        run;
    %end;
```

```

        proc append base=all data=tempfile;
            run;
        %end;
    %mend sd;

%sd;

data summer.BBand;
    set all;
run;

```

The following program evaluates the variable and fixed Bollinger Bands:

Program 3B:

```

data BB;
    set summer.finalBB;
    upperband = MA20+2*sd20;
    lowerband = MA20-2*sd20;
    upperbandl = MA20+1*sd20;
    lowerbandl = MA20-1*sd20;
    if adjprc>upperband then status = -1;
    else if adjprc<lowerband then status = 1;
    else status = 0;
run;

data VBBB;
    set BB;
    by cusip;
    drop status;
    retain period 0;
    if first.cusip then period = 0;
    if adjprc<lowerband then period = 1;
    if adjprc>lowerbandl then period = 0;
    RealPeriod = lag(period);
    if RealPeriod = 0 or RealPeriod = . then delete;
run;

data VBBS;
    set BB;
    by cusip;
    drop status;
    retain period 0;
    if first.cusip then period = 0;
    if adjprc>upperband then period = 1;
    if adjprc<upperbandl then period = 0;
    RealPeriod = lag(period);
    if RealPeriod = 0 or RealPeriod = . then delete;
run;

proc means data=VBBB;
    var ret;
run;

proc means data=VBBS;
    var ret;
run;

```

SAS Output

The MEANS Procedure

Analysis Variable : RET Returns				
N	Mean	Std Dev	Minimum	Maximum
397260	0.000969379	0.0304855	-0.6041667	8.2500000

The MEANS Procedure				
Analysis Variable : RET Returns				
N	Mean	Std Dev	Minimum	Maximum
531394	0.000335484	0.0204059	-0.3385417	0.5217391

We can see from the above output that an investor can earn 0.0969379% at the buy side. In this instance, the investor's profit is higher than when utilizing the buy-and-hold strategy.

CONCLUSION

We tested three major technical analytical strategies: moving average, RSI, and BOLL. As we mentioned above, the essential concepts of these three strategies are quite different. According to our research, BOLL strategy is not efficient; the variable-length moving average (VMA) is efficient; both the fixed-length and variable-length RSI are also efficient. Furthermore, the result of VMA is more efficient than FRSI and/or VRSI if we focus on the significance of the average daily return. One reason is that some VMA average daily returns are higher than the returns of RSI. When utilizing the VMA (1-day/50-day), its average daily return of buy is above 0.4% and the return of sell is also near -0.4%. Another reason is that the transaction days (N(Buys) and N(Sell)) are more than the ones in RSI. It states that the investor has longer time to enjoy the higher returns utilizing the VMA strategy. As a result, the variable-length moving average (VMA) is the best of the technical strategies in our research.

Generally speaking, during the period from 1981 to 2010, investors could make more money using specific technical trading rules.

CONTACT INFORMATION

Wei Wang
 University of Arizona
 Eller College of Management ,
 McClelland Hall 417, 1130 E. Helen St.
 Tucson, AZ 85721-0108
 E-mail: weiwang33@email.arizona.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration. Other brand and product names are trademarks of their respective companies.