CS7646 Project 6: Indicator Evaluation

Thomas Kim
tkim338@gatech.edu

1 INDICATORS

1.1 SMA (simple moving average)

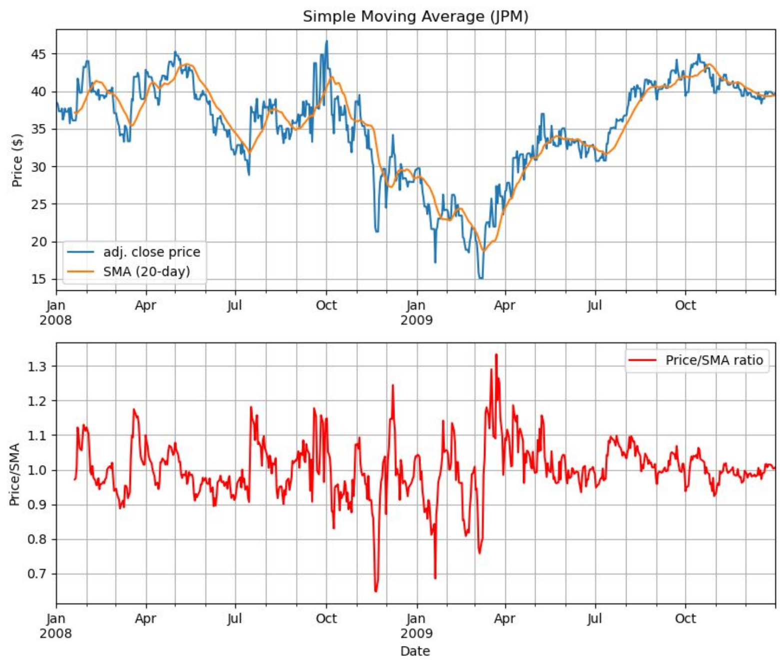

Figure 1—SMA (simple moving average) plot against adjusted close price of JPM stock.

A simple moving average, or SMA, is just the mean stock price across a given rolling window (Murphy, 1999) (pseudocode below). In the figure above, this rolling window stretches a span of 20 days, so the first 20 days of the SMA plot is empty and the first data point sits at the average stock price of the first 20 days of JPM stock.

```
sma = []
for each window in stock_price:
    compute mean
    sma.append(mean)
return sma
```

Figure 2—SMA pseudocode.

SMA is essentially a lagging indicator of the stock price minus short-term volatility ("short-term" meaning spans of time smaller than the rolling window size). This means that if the stock price deviates from the SMA, it will always eventually return, whether by deviating for a period of time longer than the SMA rolling window and causing SMA to move correspondingly, or by returning to SMA after a period of temporary volatility has passed. Therefore, buy signals can be generated whenever stock price is lower than SMA and sell signals can be generated whenever stock price is higher than SMA. However, this will not work when the stock price steadily moves up or down, in which case, SMA will always lag behind, which won't provide a useful indicator for producing a positive return.

1.2 Bollinger Band

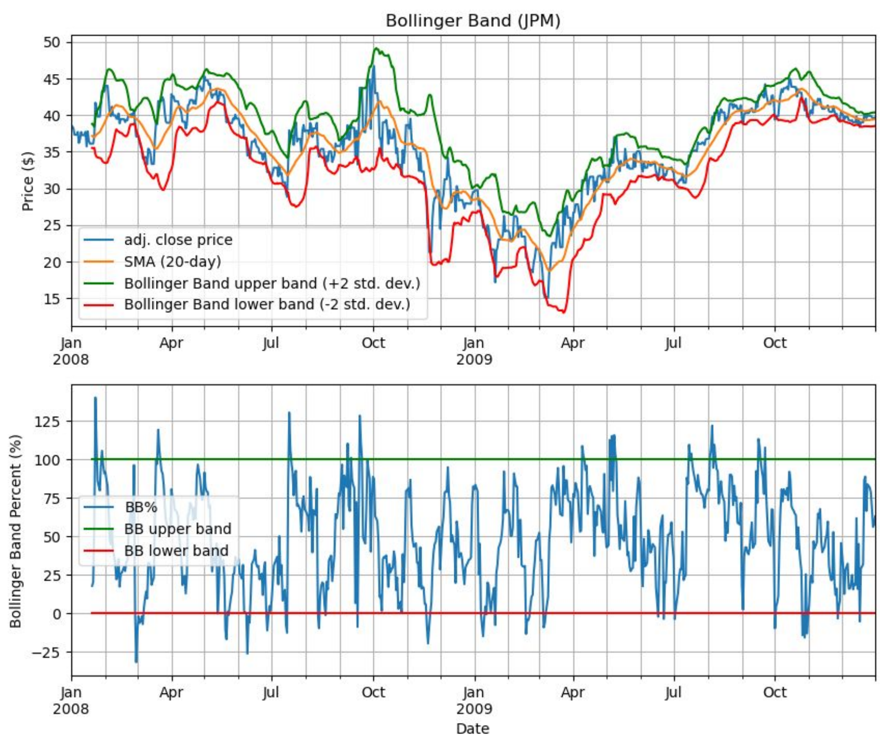

Figure 3—Bollinger Band plot using a 20-day SMA and +/- 2 standard deviation upper and lower bands overlaying JPM stock price.

A Bollinger Band is modification of an SMA with the addition of upper and lower bounds separated from the SMA by a number of standard deviations of the stock price (Bollinger, 2002), usually +/- 2 standard deviations, as was used in the figure above.

```
bollinger_lower = []
bollinger_upper = []
for each window in stock_price:
    compute mean
    compute std_dev
    bollinger_lower.append(mean - std_dev)
    bollinger_upper.append(mean + std_dev)
return bollinger_lower, bollinger_upper
```

Figure 4—Bollinger Band pseudocode.

The usage of Bollinger Bands as indicators is similar to that of just SMA, where a stock price can be assumed to return to SMA when it deviates. The application of standard deviation bounds to Bollinger Bands eliminates the issue present when using SMA as an indicator in cases where stock price steadily increases or decreases and SMA lags behind. Buy signals are generated when stock price enters the Bollinger Band from below the lower bound and sell signals are generated when stock price enters the Bollinger Band from above the upper bound. Using standard deviation here instead of SMA as used previously eliminates some cases of a lagging indicator when stock price changes steadily, as standard deviation will shrink and reduce instances of inaccurate, lagging signals.

1.3 Momentum

Momentum is an indicator that illustrates how quickly a stock price changes. Mathematically, it is simply the derivative of stock price. The implementation of this indicator is shown below in pseudocode.

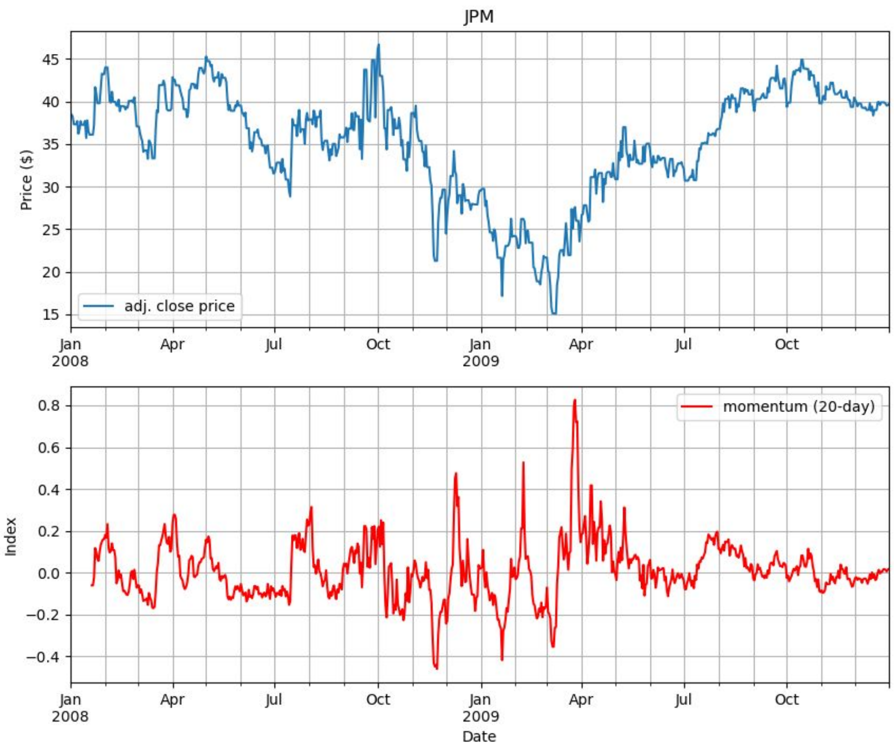

Figure 5—Plot of momentum using a 20-day rolling window against JPM stock price.

```
momentum = []
for each window in stock_price:
    m = window[i_last]/window[i_first] - 1
    momentum.append(m)
return momentum
```

Figure 6—Momentum pseudocode.

Momentum shows the rate of change of stock prices (Murphy, 1999), so to use it as an indicator presumes that other stock market participants are subject to the hot hand fallacy (Gilovich, Vallone, & Tversky, 1985), a cognitive social bias in which people tend to believe that a stock price that is increasing is more likely to keep increasing rather than decrease. A momentum indicator can be used to get ahead of this trend, generating buy signals when momentum is high and sell signals when momentum is low.

1.4 Volatility

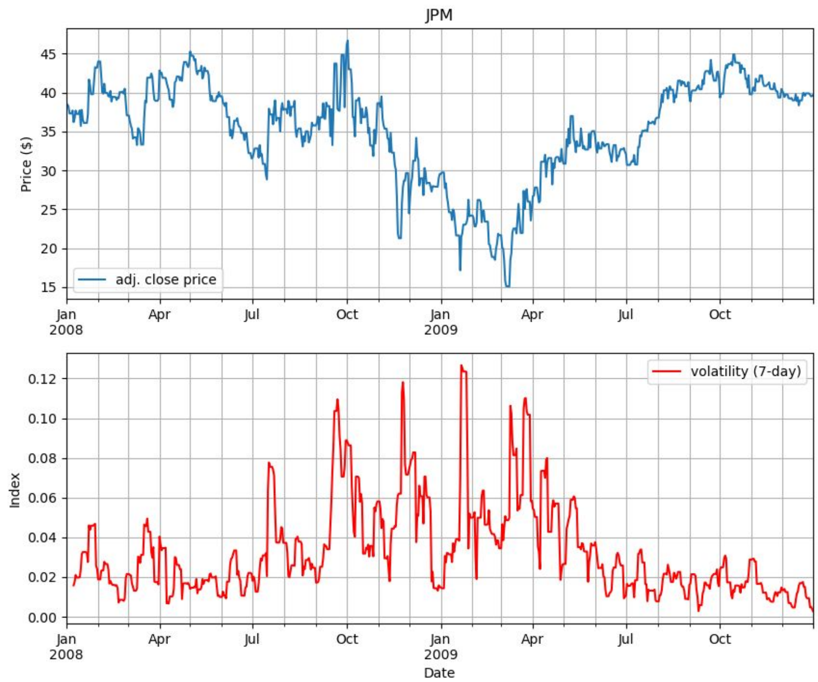

Figure 7—Plot of volatility using a 7-day rolling window against JPM stock price.

Volatility is a measure of how disperse a stock price is over a given time period (Murphy, 1999). Mathematically, this is calculated as the standard deviation of daily returns across a given window of time. Pseudocode for this implementation is shown below.

```
volatility = []
for each window in daily_return:
    compute std_dev
    momentum.append(std_dev)
return momentum
```

Figure 8—Volatility pseudocode.

The fact that volatility is a measure of standard deviation of returns generally means that high volatility goes hand-in-hand with high risk, and is thus used as an indicator of risk. Volatility alone does not generate buy/sell signals for traditional trading (with the exception of implied volatility speculation and options trading) and can instead be used to evaluate risk and develop trades with better risk/reward ratios and to hedge risk. For simple long/short trading, high volatility can be used as an indicator to avoid trading a stock entirely while low volatility can be used as an indicator that trading a stock is less risky.

1.5 OBV (on-balance volume)

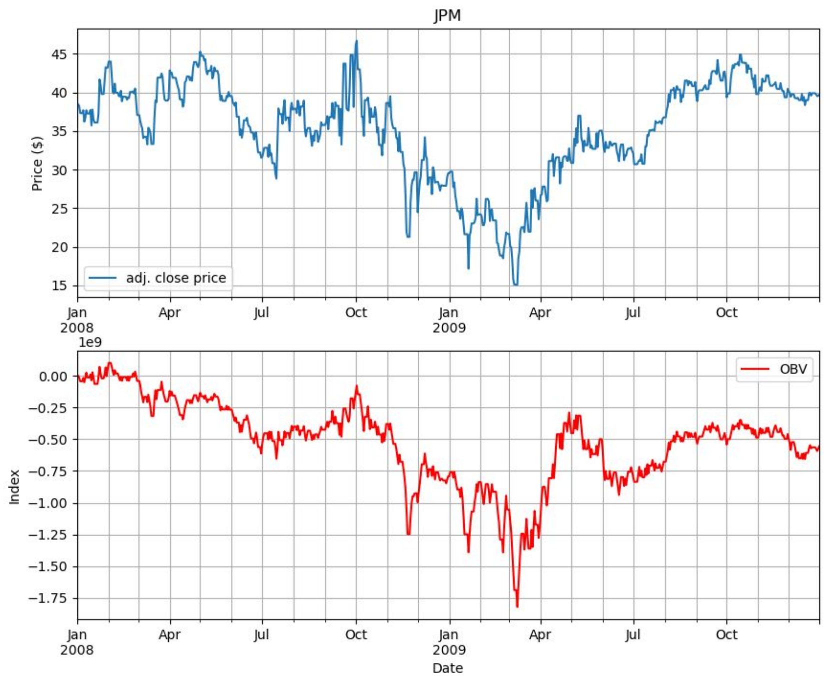

Figure 9—Plot of OBV (on-balance volume) against JPM stock price.

On-balance volume (OBV) is a measure of volume flow related to stock price (Murphy, 1999). A positive daily return leads to that day's trading volume to be added to current OBV while a negative daily return leads to that day's trading volume to be subtracted from current OBV. Pseudocode implementation of OBV is shown below.

```
obv = []
for each price in stock_price:
    if daily_return == 0:
        # do nothing
    elif daily_return > 0:
        curr_obv += trade_volume
    else:
        curr_obv -= trade_volume
    obv.append(curr_obv)
return obv
```

Figure 10—OBV pseudocode.

OBV is believed to illustrate the distinction between actions of institutional investors and retail investors. Large funds may buy large volumes of a stock before the price begins to increase, as the lowest ask orders are filled first. The inverse is also true, where large funds may sell large volumes of a stock before the price of the stock decreases. Buy signals can be generated when OBV is high and sell signals can be generated when OBV is low.

2 THEORETICALLY OPTIMAL STRATEGY

2.1 Implementation

The theoretically optimal strategy was implemented by determining the price change from day to day and obtaining the portfolio position with the best return for each price change. For this implementation, only one trade per day was permitted, as the required output had to contain only daily trades. Adjusted close price was thus used to determine price changes.

Beginning with the start date, stock price for the current day was obtained and compared with the following day's price. If the price was higher on the following day, the portfolio's stock position was set to +1000 shares (long). If the price was lower on the following day, the portfolio's stock position was set to-1000 shares (short). The portfolio's current position was maintained in order to compute the necessary trade required to reach the desired position. This trade was stored along with the date of the trade and these data were compiled into a trades dataframe that was then outputted.

Days where a stock price was returned as a nan value were assumed to not be trading days and were ignored, skipping over them to the next valid trading day.

2.2 Assumptions

Notable specifications explicitly given in the assignment instructions were that zero commission and zero impact were to be used, along with unlimited leverage but a limited stock holding of -1000 to +1000 shares.

It was assumed (or more accurately, inferred) that trades for the theoretically optimal strategy were limited to one per day, as the required output allowed for one trade per date. It was also assumed that every trade would be completely filled at the given stock price, every day. For 1000 stocks for a high volume stock symbol, this may be reasonable, but this may not be the case for low volume stocks. Another assumption was that when a stock price was available for the given symbol (JPM), that day was a trading day and trades could be made.

2.3 Results

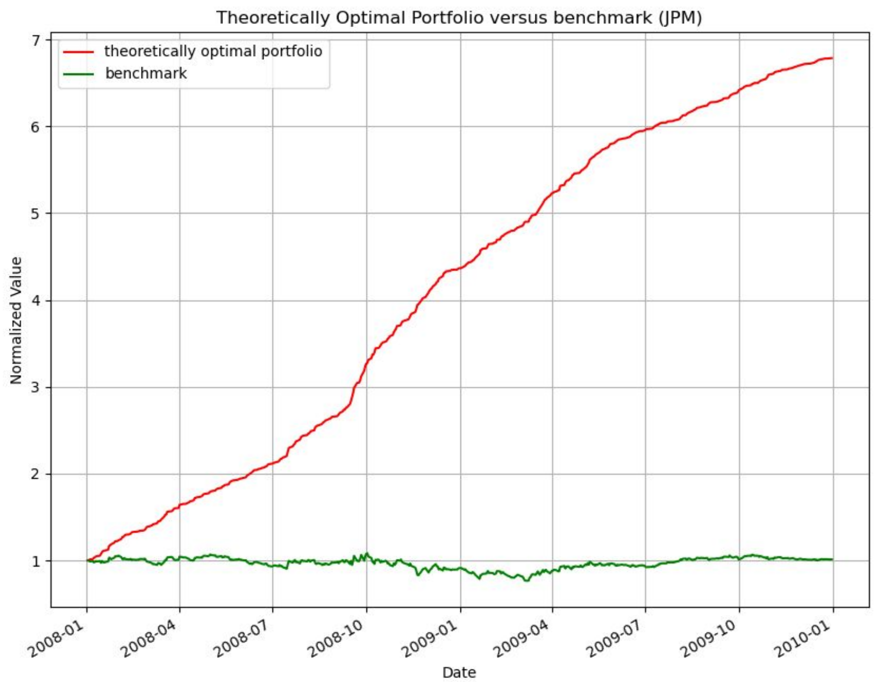

Figure 11—Performance of the theoretically optimal portfolio against a benchmark.

Table 1—Performance metrics of the theoretically optimal portfolio compared to a benchmark.

	Theoretically optimal portfolio	**Benchmark**
Cumulative return	5.786	0.01230
Standard deviation of daily returns	0.004542	0.01697
Mean of daily returns	0.003809	0.0001678

3 REFERENCES

1. Murphy, J. (1999). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. In New York Institute of Finance.
2. Bollinger, J. (2002). Bollinger on Bollinger Bands. McGraw Hill.
3. Gilovich, T., Vallone, R., & Tversky, A. (1985). The Hot Hand in Basketball: On the Misperception of Random Sequences. In Cognitive Psychology.