Exploring Variance-Based Alternatives to the Golden Crossover Strategy

Summary

The Golden Cross-over strategy is a popular technical analysis trading strategy that involves the crossing over of two moving averages, typically the 50-day moving average and the 200-day moving average. In this paper, we provide an overview of the history and creation of the Golden Cross-over strategy, as well as its use and overall performance. We also expand on the different windows of time used in this strategy and explain why the 200-50 window is the most popular.

Furthermore, we investigate how to use a stock's historical volatility to create dynamic trading windows for the Golden Cross-over strategy. By analyzing historical data and determining the optimal moving averages that capture the stock's volatility and trends, we define the rules for entry and exit. Our research shows that using dynamic windows based on historical volatility can improve the performance of the Golden Cross-over strategy by providing faster or slower signals depending on the stock's characteristics.

Overall, our findings suggest that adapting the Golden Cross-over strategy using historical volatility for dynamic trading windows can lead to improved trading results by optimizing the strategy based on changing market conditions and the stock's volatility.

On the Golden Cross-Over Strategy

The Strategy

The Golden Cross-over strategy is a popular technical analysis trading strategy used by traders to identify potential changes in market trends. This strategy involves the crossing over of two moving averages, typically the 50-day moving average and the 200-day moving average. When the 50-day moving average crosses above the 200-day moving average, it is considered a bullish signal, indicating a potential upward trend in the market. Conversely, when the 50-day moving average crosses below the 200-day moving average, it is considered a bearish signal, indicating a potential downward trend in the market.

The Golden Cross-over strategy has its roots in the Dow Theory, a theory of market analysis developed by Charles Dow in the late 19th century. The idea behind the strategy is to capture the short-term and long-term trends in the market by using two moving averages that are widely recognized by traders.

The Problem - Choosing Ideal Time Windows

The windows of time used in the Golden Cross-over strategy may vary depending on the trader's preference and the specific financial instrument being traded. However, the most commonly used windows of time for the Golden Cross-over strategy are:

- 50-day and 200-day moving averages: This is the most popular combination used for the Golden Cross-over strategy. It is widely used in the stock market and other financial instruments.
- 20-day and 50-day moving averages: This combination is used by traders who prefer shorter-term signals.

- 100-day and 200-day moving averages: This combination is used by traders who prefer longer-term signals.

Deciding which window to choose comes with tradeoffs: systems that prefer shorter-term signals are prone to falling for false signals, while systems that prefer longer-term signals may miss good opportunities. The 200/50 window is the most used window for the Golden Cross-over strategy because it is a good midpoint between long and short-term signals, and thus it became a widely recognized and established standard in the industry. Additionally, the 200-day moving average is considered a key support or resistance level, making it a significant level for traders to watch. The 50-day moving average provides a shorter-term signal that can confirm the trend reversal signaled by crossing the 200-day moving average.

However, the widows that provide the optimal balance between short and long-term signals are unlikely the same across different stocks, and approaching this optimal window may lead to improved returns.

The Solution - Variance-based Time Windows

For stocks whose intrinsic volatility is high, shorter-term signals may pose an increased risk to fall for false signals. On the other hand, more stable stocks, if paired with longer-term time windows, will likely miss opportunities it will take longer for their moving averages to cross.

To account for that, we developed variance-based time windows: we increase the longer time window based on the stock's volatility in order to make highly volatile stocks less sensitive to smaller changes, and more volatile stocks more sensitive to changes.

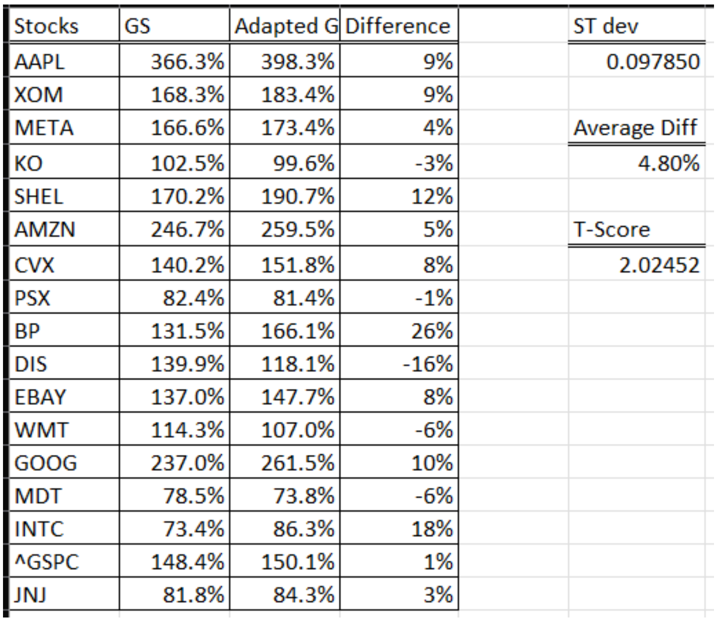
The Method

First, we adopted Python as our standard work tool. Furthermore, in our code, based on a given ticker and start date we extract quotes from Yahoo and proceed to calculate the stock's historical volatility as a proxy for its intrinsic volatility.

Moreover, we developed the code that performs the standard Golden Cross-over strategy by setting short/long signals that change when the moving averages cross each other. We set the standard method to use the regular 200/50 windows.

We proceed to develop our variance-based time windows. We plan it in a way that the short window remains at 50 days, while the long window becomes an increasing function of the stock's volatility. Using the approximate value of 20% for the market volatility, we set the long window as $166 * (1 + volatility)$, so that a stock whose volatility is 20% will have the regular 200-day long average window. Finally, we calculate cumulative gains and plot a comparison of both methods.

Results and Next Steps



Stocks	GS	Adapted G	Difference		ST dev
AAPL	366.3%	398.3%	9%		0.097850
XOM	168.3%	183.4%	9%		
META	166.6%	173.4%	4%		Average Diff
KO	102.5%	99.6%	-3%		4.80%
SHEL	170.2%	190.7%	12%		
AMZN	246.7%	259.5%	5%		T-Score
CVX	140.2%	151.8%	8%		2.02452
PSX	82.4%	81.4%	-1%		
BP	131.5%	166.1%	26%		
DIS	139.9%	118.1%	-16%		
EBAY	137.0%	147.7%	8%		
WMT	114.3%	107.0%	-6%		
GOOG	237.0%	261.5%	10%		
MDT	78.5%	73.8%	-6%		
INTC	73.4%	86.3%	18%		
^GSPC	148.4%	150.1%	1%		
JNJ	81.8%	84.3%	3%		

Based on a diverse portfolio of 17 stocks, we observe that our altered Golden Crossover strategy shows statistically significant improvement on a confidence interval of 95%

*Quotes used:

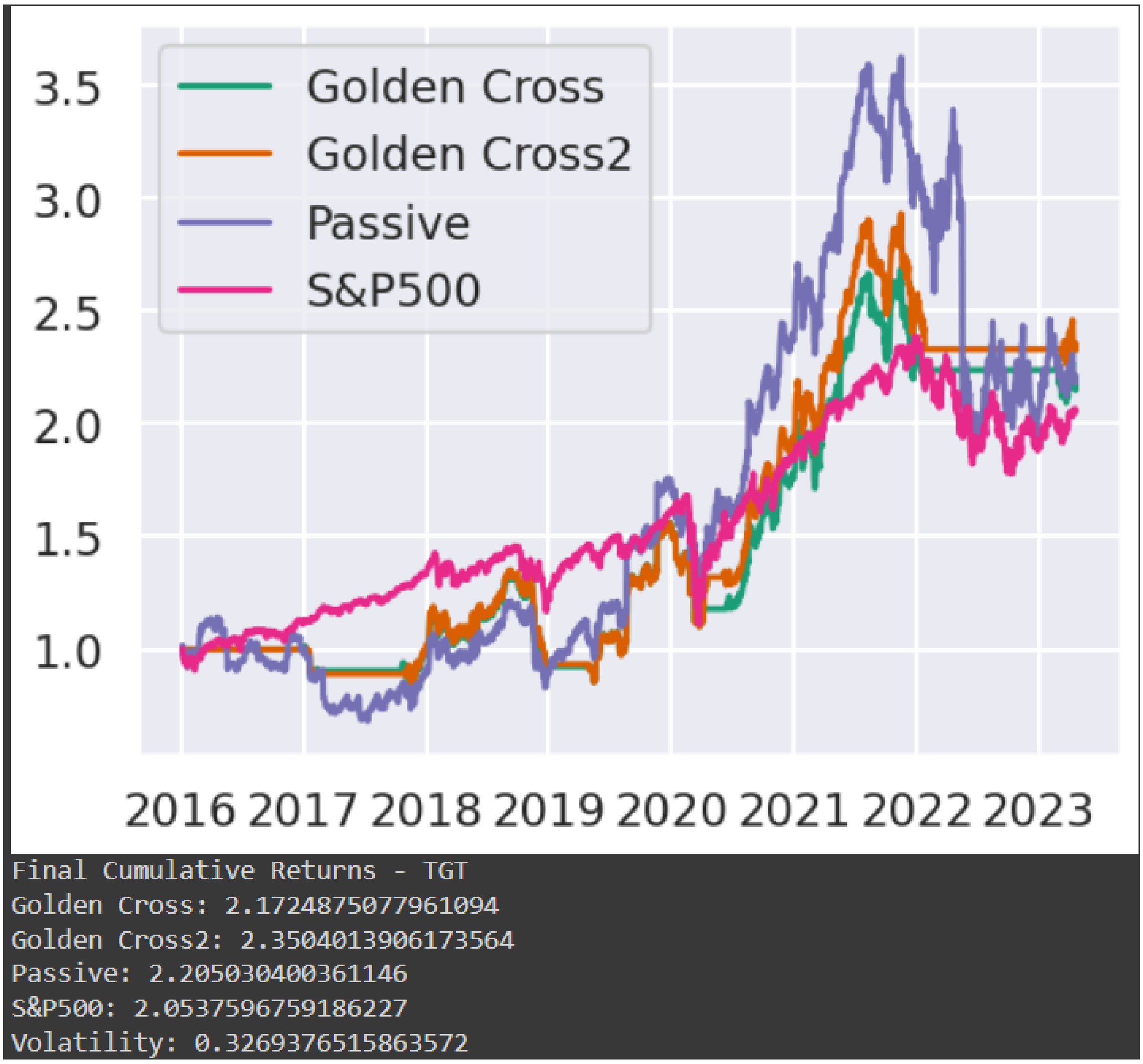
From 2016-01-01 to 2023-04-20

To further progress with our method, our next steps include:

- Including analysis of information density to give more or less weight to recent data
- Fine-tuning variance-based windows
    - Testing different ways to implement variance in the time-windows and comparing their relative performances
- Performing stress tests to find how different categories of stock perform with the model

Additional Images and Code

Results for Target Corp stock (TGT). In this case, the adapted Golden Cross strategy outperformed not only the standard Golden Cross but also the passive strategy



```
Final Cumulative Returns - TGT
Golden Cross: 2.1724875077961094
Golden Cross2: 2.3504013906173564
Passive: 2.205030400361146
S&P500: 2.0537596759186227
Volatility: 0.3269376515863572
```

Code:

```
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np

def hVol (data):
    log_returns = np.log(data['Close'] / data['Close'].shift(1))
    volatility = log_returns.std() * np.sqrt(252)
    return volatility
```

```
# Define the ticker symbol and time window for the golden cross
ticker = "^GSPC"

startD = "2016-01-01"

# Get historical market data for the ticker symbol
stock_data = yf.download(ticker, start=startD)

vol = 1+hVol(stock_data)
golden_cross_short_window = 50
golden_cross_long_window = 200
shortW = round(50*1)
longW = round(166*vol)

# Calculate the moving averages for the golden cross
stock_data['SMA_short'] =
stock_data['Close'].rolling(window=golden_cross_short_window).mean()
stock_data['SMA_long'] =
stock_data['Close'].rolling(window=golden_cross_long_window).mean()


# Generate signals for the golden cross 1
stock_data['Signal'] = 0.0
stock_data['Signal'] = np.where(stock_data['SMA_short'] >
stock_data['SMA_long'], 1.0, 0.0)


# Calculate the daily returns for the stock
stock_data['Returns'] = stock_data['Close'].pct_change()

# Calculate the cumulative returns for the golden cross strategy 1
stock_data['Strategy Returns'] = stock_data['Signal'].shift(1) *
stock_data['Returns']


# Calculate the cumulative returns for the passive strategy
stock_data['Passive Returns'] = stock_data['Returns']
stock_data['Passive Cumulative Returns'] = (1 + stock_data['Passive
Returns']).cumprod()

# Calculate the cumulative returns for the S&P500 index
sp500_data = yf.download('^GSPC', start=startD)
```

```
sp500_data['Returns'] = sp500_data['Close'].pct_change()
sp500_data['Cumulative Returns'] = (1 +
sp500_data['Returns']).cumprod()

# Priority change to recent data for overperforming stockv- On test,
not part of main project
#mktD = stock_data['Passive Cumulative Returns'][-1]-
sp500_data['Cumulative Returns'][-1]
#print(mktD)
#shortW = round(50*1/mktD/2)
#longW = round(166*vol/mktD/2)

# Calculate the moving averages for the golden cross 2
stock_data['SMA_short2'] =
stock_data['Close'].rolling(window=shortW).mean()
stock_data['SMA_long2'] =
stock_data['Close'].rolling(window=longW).mean()

# Generate signals for the golden cross 2
stock_data['Signal2'] = 0.0
stock_data['Signal2'] = np.where(stock_data['SMA_short2'] >
stock_data['SMA_long2'], 1.0, 0.0)

# Calculate the cumulative returns for the golden cross strategy 2
stock_data['Strategy Returns2'] = stock_data['Signal2'].shift(1) *
stock_data['Returns']

# Calculate the cumulative returns for the stock, golden cross, and
passive strategies
stock_data['Cumulative Returns'] = (1 + stock_data['Strategy
Returns']).cumprod()
stock_data['Passive Cumulative Returns'] = (1 + stock_data['Passive
Returns']).cumprod()
stock_data['Cumulative Returns2'] = (1 + stock_data['Strategy
Returns2']).cumprod()

# Plot the cumulative returns for the stock, golden cross, passive, and
S&P500 strategies
plt.plot(stock_data.index, stock_data['Cumulative Returns'],
label='Golden Cross')
plt.plot(stock_data.index, stock_data['Cumulative Returns2'],
label='Golden Cross2')
```

```
plt.plot(stock_data.index, stock_data['Passive Cumulative Returns'],
label='Passive')
plt.plot(sp500_data.index, sp500_data['Cumulative Returns'],
label='S&P500')
plt.legend(loc='upper left')
plt.show()

# Print the final cumulative returns for the stock, golden cross,
passive, and S&P500 strategies
print('Final Cumulative Returns -', ticker)
print('Golden Cross:', stock_data['Cumulative Returns'][-1])
print('Golden Cross2:', stock_data['Cumulative Returns2'][-1])
print('Passive:', stock_data['Passive Cumulative Returns'][-1])
print('S&P500:', sp500_data['Cumulative Returns'][-1])
print('Volatility:', hVol(stock_data))
```