Predicting Lagging Asset Volatility in Lead-Lag Asset Pairs

Aarush Chaubey, Asim Ali, William Xu, Annie Yao

December 2024

Abstract

When pricing European option using the Black-Scholes model, it is often not possible to know the exact implied volatility of the asset. Thus, in practice, quants have resorted to other methods: deriving volatility from other options on the same asset, extrapolating using historical volatility, or using GARCH models to estimate volatility. In this project, we derive a strategy to dynamically estimate the volatility of the lagging asset in a highly correlated time-delayed pair.

1 Introduction

When trying to price an option, the difficulty of using the Black-Scholes model is in obtaining an accurate measure of implied volatility. In our example, we will assume that we have

- r, the risk free rate of return,
- S, the current price of the asset,
- K, the strike price of the option,
- T, the time to expiration,

and our goal is to estimate said volatility, denoted σ.

Suppose assets A and B are highly correlated with a lead-lag relationship such that a change in the price of the leading asset, A, drives changes in the price of lagging asset, B, τ time later. We assume that change in volatility of A, denoted σ_A, also implies a change in volatility of the B, denoted σ_B, after a delay—thus, we will use the former to predict the latter. However, due to the time delay, liquidity, and other conditions specific to each asset, the relationship between σ_A and σ_B may have too much noise to reliably predict one from the other directly.

To account for this, we use a ternary search to find the value τ that maximizes the correlation, ρ, of S_A and S_B over a given window of time Θ. τ should represent the delay both in the change in the stock prices, as well as the volatilities.

Thus, we can perform a linear regression on σ_A and σ_B after matching each data point σ_B to one τ time earlier, σ_{A^τ}, to obtain the equation $\sigma_B = \mu\sigma_{A^\tau} + \beta$.

Now, we can use this regression to forecast σ_B from σ_{A^τ}.

2 Dataset

For our data, we chose two stocks that we believe to be highly correlated and to have a lead-lag relationship, being Cleveland-Cliffs (CLF) and Nucor (NUE). All of the data of the stock was obtained using yfinance, a Python library that downloads market data directly from Yahoo! Finance's API. All of the data is from the most recent month, and limited to 1-minute intervals at the shortest.

3 Methods

3.1 Searching for τ

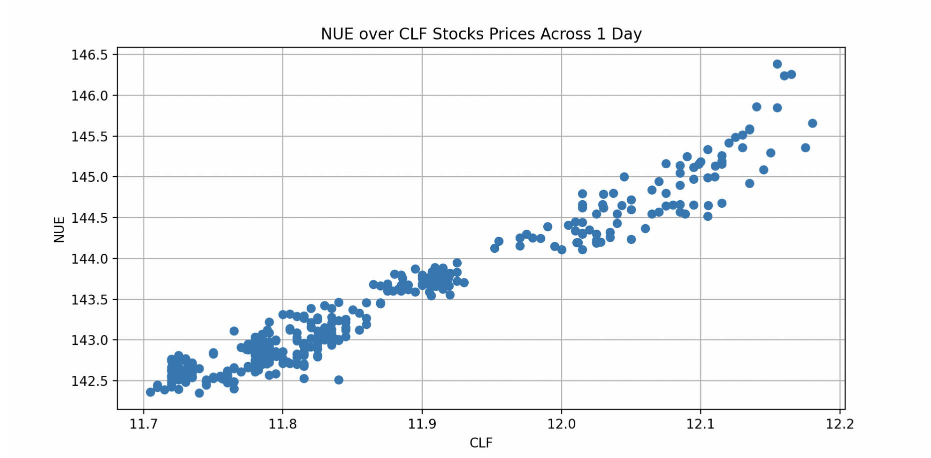
To optimize for τ, we compare the historical stock data for S_1, S_2 over a certain time frame $\Theta = [t_1, t_2]$ such that all pairs $(t_1, t_2) \in \mathbb{N}^+ \times \mathbb{N}^+$. Adding τ to this time frame shifts Θ to be $[t_1 + \tau, t_2 + \tau]$.

We define a correlation function $C(S_1(\Theta+t), S_2(\Theta))$, where the leading stock is t minutes ahead of the lagging stock. Effectively, our claim is that τ maximizes C.

Because of this claim, we note that $\tau' < \tau \to C(\tau') < C(\tau)$ Furthermore $\tau' > \tau \to C(\tau') < C(\tau)$. Thus, $C(\tau')$ is concave over the interval we are searching, so we can use ternary search to maximize C.

```
def ternary_search(l, r):
    eps = 600 #predefined value
    while r - l > eps:
        m1 = l + (r - l) // 3
        m2 = r - (r - l) // 3
        print(m1, m2, l, r)
        f1 = C(m1)
        f2 = C(m2)
        # print(f1, f2)
        if f1 < f2:
            l = m1
        else:
            r = m2
    return (l+r)//2 #final-correlation=0.926
```

After adjusting for tau, our data looks like this.

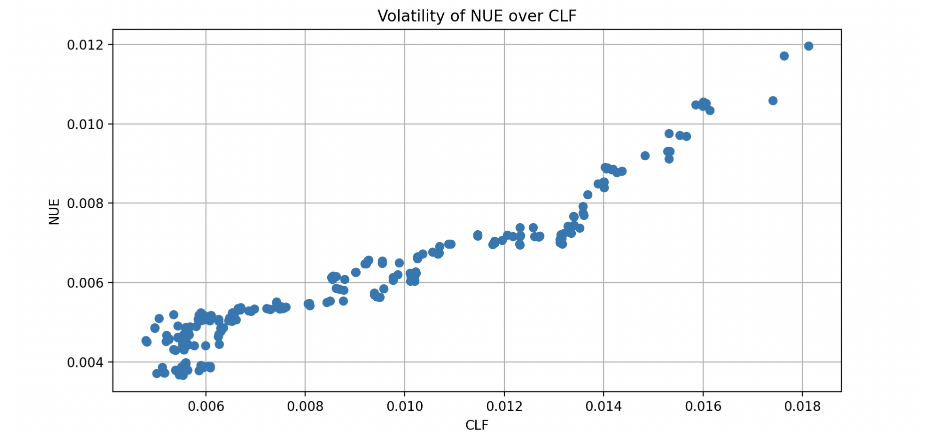


3.2 Defining C

After shifting the leading asset by τ', we take the correlation of corresponding pairs of elements in the shifted S_1 and S_2. Because there exist values in both lists for which there are no corresponding values, we remove all such values. After this, we calculate the correlation using NumPy.

3.3 Calculate Historical Volatility

We now have to find the scale factor ρ such that $\sigma_1' = \rho\sigma_2'$ where σ_1' is calculated over $\Theta + \tau$ and σ_2' is calculated over Θ. To do this, we use the same lists as in 3.2 and calculate their respective volatility arrays using a historical volatility model. This model involves calculating a rolling standard deviation over a window size of 60 minutes.

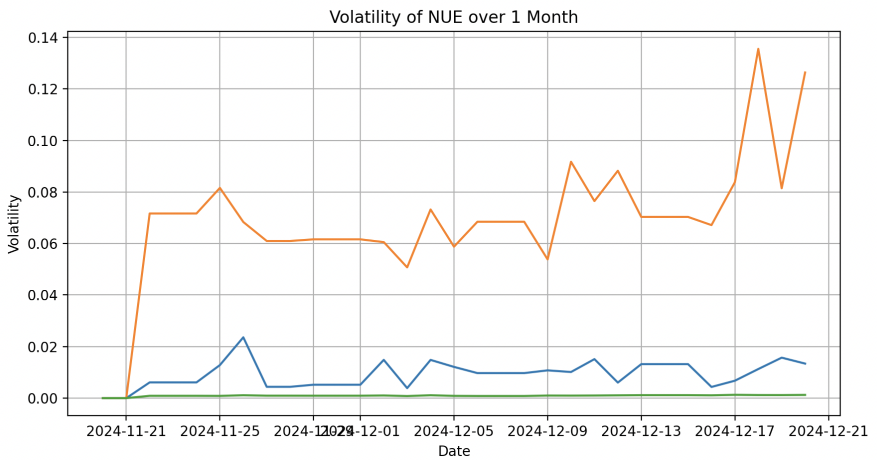


Then, we use Scikit to linearly regress these volatilities, leading to $\mu = 0.76505$ and $\beta = -7.788 \cdot 10^{-4}$.

4 Results

To test our model, we used Cleveland Cliffs (CLF) as our leading asset and Nucor (NUE) as our lagging asset, and we considered the value of their stocks in 1-minute intervals over a 7-day period. As we were not able to acquire the historical data for the options, we are unable to compare our calculated volatility with the actual implied volatility based on the market price of the options. However, we are able to compare the value of μ (the ratio of σ_1 to σ_2) across multiple different models.

In the following graph, the green line represents historical volatility, blue represents our correlated volatility, and orange represents the GARCH forecasted volatility.



1. GARCH Predicted Volatility: 0.06058
2. Our Predicted Volatility: 0.01461
3. Historical Volatility: 0.00102

Because of its low order of magnitude, the historical volatility is not able to accurately predict the lagging asset's volatility. However our predicted volatility is the same order of magnitude as GARCH, providing some credibility of accuracy.

5 Limitations

In very liquid markets, visible lags are nearly immediately corrected for by the market. The smallest interval of time the API from which we pulled our data from is able to offer is one minute, which means that any lag will usually be correct for well under the interval. Even in less liquid markets or scenarios where there may exist consistent time delay (e.g. large-cap and small-cap stocks in the same industry), the 1-minute interval is still too wide. Ideally, we would be able to backtest on data with, at the very least, 1-second intervals.

Furthermore, even though the outset of our strategy was to calculate volatility for option pricing with Black-Scholes, we were unable to source historical data on the options of our assets. Instead, the volatility data used for the linear regression was directly calculated from the stock data. Thus, there may have been a discrepancies between the computed data and the actual implied volatility obtained from the market price of the options of the assets.

This also meant that we were unable to compare the results to actual historical data to evaluate its accuracy. We could only compare it to other volatility forecasting models, which, based on our dataset, may not be as accurate of a metric.

Possible ways to improve the accuracy of our model include potentially dynamically recalculating τ over shorter periods of time. In some cases, our volatilities over the 60-minute time window were very highly linearly correlated, but in other cases the correlation was less strong or less linear—possibly indicating that using one value of τ across the whole day may be inaccurate.