

WiDS Assignment -2: Kalman Filter and Backtesting in Pairs Trading

Mentors: Nimay Shah, Devavrat Patni

December 11, 2024

This assignment builds upon Assignment-1, where you implemented a Pairs Trading strategy using Machine Learning (ML). In this continuation, you will use Kalman Filtering to dynamically estimate the relationship between two stocks and conduct rigorous backtesting to evaluate the strategy's performance. The focus will be on Adobe Inc. (ADBE) and Microsoft Corp. (MSFT).

1 Problem Statement

The task is to implement a Pairs Trading strategy using Kalman Filtering and perform detailed backtesting to evaluate its performance.

Key Objectives

1. Use Kalman Filtering to dynamically estimate the time-varying relationship between ADBE and MSFT.
2. Perform backtesting over a historical dataset to evaluate strategy performance.
3. Generate buy/sell signals and calculate cumulative profit and loss (PnL).

Step 1: Fetch Historical Stock Data

Obtain the stock price data for ADBE and MSFT from Yahoo Finance for a defined time period.

Step 2: Kalman Filtering

1. Model the dynamic relationship between ADBE and MSFT using the Kalman Filter.
2. Estimate the parameters α_t and β_t in the linear regression model:

$$y_t = \alpha_t + \beta_t x_t + \epsilon_t, \tag{1}$$

where x_t is the price of ADBE and y_t is the price of MSFT.

Step 3: Generate Trading Signals

1. Define the spread as:

$$s_t = y_t - (\alpha_t + \beta_t x_t). \quad (2)$$

2. Generate buy/sell signals based on deviations of s_t from its mean:

- **Buy Signal:** s_t is significantly below its mean.

- **Sell Signal:** s_t is significantly above its mean.

Note: While you can use a simple strategy to find the difference between the spread value and the spread mean, and thresholding it to generate the trading signal, a more robust idea would be finding how many standard deviations (rolling) is the spread value away from the mean.

Step 4: Backtesting

1. Simulate the trading strategy over the historical dataset using the generated signals.
2. Calculate profit and loss (PnL) for each trade.
3. Evaluate overall strategy performance by plotting cumulative PnL.

Hints

1. Use the Python library `pykalman` for Kalman Filtering.
2. Ensure your backtesting framework accounts for transaction costs.
3. Use a rolling window to calculate the spread's mean and standard deviation dynamically.

May The Profit Be With You!