Dynamic Trading Strategy Design based on PCA

Yi Liu

USC ID: 8666552780

1 Introduction of dataset

1.1 Dataset

I introduced the top 100 stocks and their daily close price for the PCA analysis, according to the rank on the website (Dec 7).[1] Since the data is available and complete before Dec.2021, I select the close price data during the period of 2019- 2020 as the in-sample data to define a trading strategy, and use data in 2021 as the out-of-sample to test that strategy.

1.2 Data Pre-processing

All stocks are accessible from Yahoo Finance, but there are quite a lot null values in the table, due to marketing closing and missing records. Although the stock list includes stocks from different countries or exchanges, US stocks count for more than 90 percent. Firstly,I drop rows in light of US's closing market timetable.[2] Secondly,'2222.SR','MPNGF' and 'PRX.VI' don't have complete information during 2019-2020, so I remove these three columns.
As for other missing records, two imputers are available to choose from: IterativeImputer and KNNImputer. The former one follows the principle of Linear Regression. For example, if 'DIS' is defined as the target variable, and other 96 stocks as features, the IterativeImputer will train a LR model based on rows where the target is not null. And then, for all rows the 'DIS' stock prices are missing, the model predicts them by passing all feature values via LR regression. The latter one follows KNN. The scikit-learn will find some rows similar to the one where target's value is null, and then the model just average the known target values and fill in the empty space. KNNImputer always get better fitting parameters, so I use the latter one.

2 Introduction of PCA model

2.1 Basic introduction

Many machine learning problems involves thousands or millions of features for each training instance. Not only do all these features make training extremely slow, but they can also make it much harder to find a good solution. This problem is often referred to as the dimensionality algorithm.

[1] **stock-list =**['AAPL','MSFT','GOOG','2222.SR','AMZN','TSLA','FB','NVDA','BRK-A','TSM','TCEHY','JPM','V','HD', '005930.KS','UNH','JNJ','LVMUY','600519.SS','WMT','PG','BABA','NSRGY','BAC','RHHBY','MA','ASML','ADBE','PFE', 'DIS','NFLX','NKE','XOM','OR.PA','TM','CRM','NVO','TMO','1398.HK','ORCL','300750.SZ','CSCO','CMCSA','KO','LLY', 'COST','ABT','AVGO','PEP','ACN','CVX','DHR','PYPL','ABBV','VZ','RELIANCE.NS','INTC','3968.HK','MPNGF','QCOM', 'WFC','MCD','HESAF','MRK','INTU','NVS','TXN','SHOP','MS','UPS','NEE','CICHY','TCS.NS','RYDAF','AZN','LOW', 'AMD','PRX.VI','LIN','T','UNP','ACGBY','SAP','SONY','KYCCF','SCHW','MDT','BHP','TMUS','RY','PM','HON', '000858.SZ','PTR','PNGAY','002594.SZ','SE','CDI.PA','BLK','UL']

[2] **Closing Time during 2019-2020:** 2019-01-01,2019-01-21,2019-02-18,2019-04-19,2019-05-27,2019-07-04,2019-09-02,2019-11-28,2019-12-25,2020-01-01,2020-01-20,2020-02-17,2020-04-10,2020-05-25,2020-07-03,2020-07-04,2020-09-07,2020-10-12,2020-11-03,2020-11-11,2020-11-26,2020-12-25

Principal Component Analysis (PCA) is by far the most popular dimensionality reduction algorithm. First it identifies the hyperplane that lies closest to the data, and then it projects the data onto it.
When choosing hyperplane, selected hyperplane should preserve the maximum amount of variance, as it will most likely lose less information than the other projections. Another way is to minimizes the mean squared distance between the original dataset and its projection onto that hyperplane.

2.2 Choosing the right number of dimension

To find the principle components of a training set, PCA has a standard matrix factorization technique called Singular Value Decomposition (SVD) that can decompose the training set matrix X into the matrix multiplication of three matrices $\mathrm{U}\Sigma V^{\mathrm{T}}$, where V contains the unit vectors that define all the principal components, as the figure 1 shows.
Before decomposing the training set, the dataset should normalized, to speed up the SVD processed.

$$\mathbf{V} = \begin{pmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ | & | & & | \end{pmatrix}$$

Figure 1: V matrix

Each column of V explains some useful information. d is defined as the number of dimensions to reduce down to, which needs to add up to a sufficiently large portion of the variance (e.g. 95 %)
In this project, I wish the model can keep at least 95 % covariance structure. There are two code options to decide the minimum number of dimensions. For instance,**d=14** can be shown from plotting (figure 2).

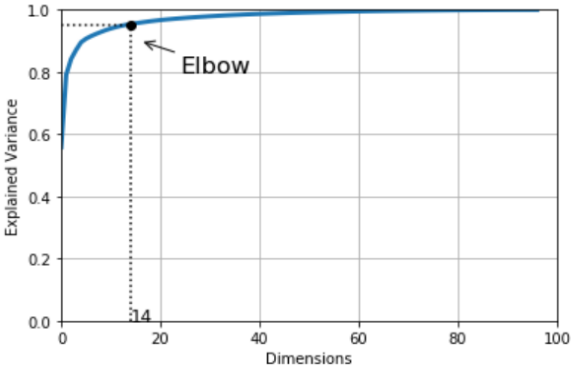
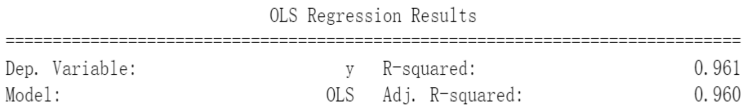


Figure 2: choosing the right number of d

3 Trading Strategy

3.1 Evaluation of model

I define Visa's close price as target, the other 96 stocks as features, and fit the data set via Linear Regression model. The R-squared of in-sample data is 0.961(figure 3), means that in 14 dimension, this model at least explains 96% of original variance. Apparently, it's a quite good performance.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.961
Model:                            OLS   Adj. R-squared:                  0.960
```

Figure 3: In-sample: OLS Linear Regression Result

3.2 Construct trading signal

When constructing trade strategy, I set the threshold as 2 standard deviation of residual (0.3948). Traders should long Visa and short the other 96 stocks multiplied by parameters when signal hits the level (mean-2std); short Visa and long the other 96 stocks multiplied by parameters when signal hits the level (mean+2std), as the figure 4 shows.
Based on the plotting of in-sample data (2019-2020), traders long Visa (short others) for 5 times, and short Visa (long others) for 5 times. Each trade earns $ 0.3948, and the total revenue is nearly $ 3.95 per share.

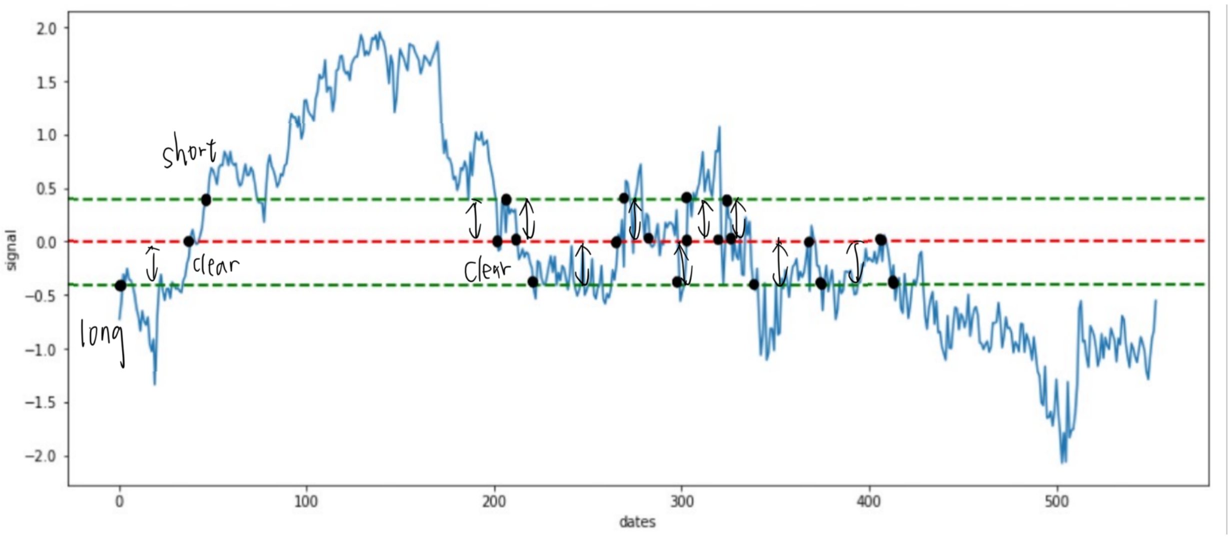
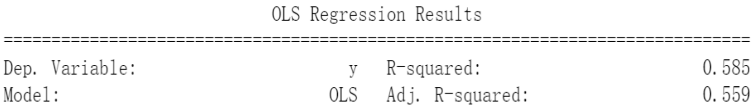


Figure 4: Trading strategy: mean ± 2std

3.3 Out of sample test

The out-of-sample ranges from 2021-01-01 to 2021-11-30. The data pre-processing includes the removal of three stocks ('2222.SR','MPNGF' and 'PRX.VI'), dropping all empty rows and dropping rows when US stock market closed. Then I also use KNNImputer to fill in all missing records.
The R-squared of out-of-sample data is 0.585 (figure 5), revealing that almost 60% visa's stock prices can be predicted by the other 96 features. Although some P values are large due to non-significant coefficient(especially the constant's P values), most of P values are zero. In all, using Linear Regression to fit the out-of-sample data is acceptable.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.585
Model:                            OLS   Adj. R-squared:                  0.559
```

Figure 5: Out-of-sample: OLS Linear Regression Result

When executing trading strategy on out-of-sample dataset(2021), traders long Visa (short other 96 stocks) for 37 times, short Visa (long others) for 41 times, each trade earns \$ 0.3948. So, traders will get \$ 30.79 per share (Figure 6).

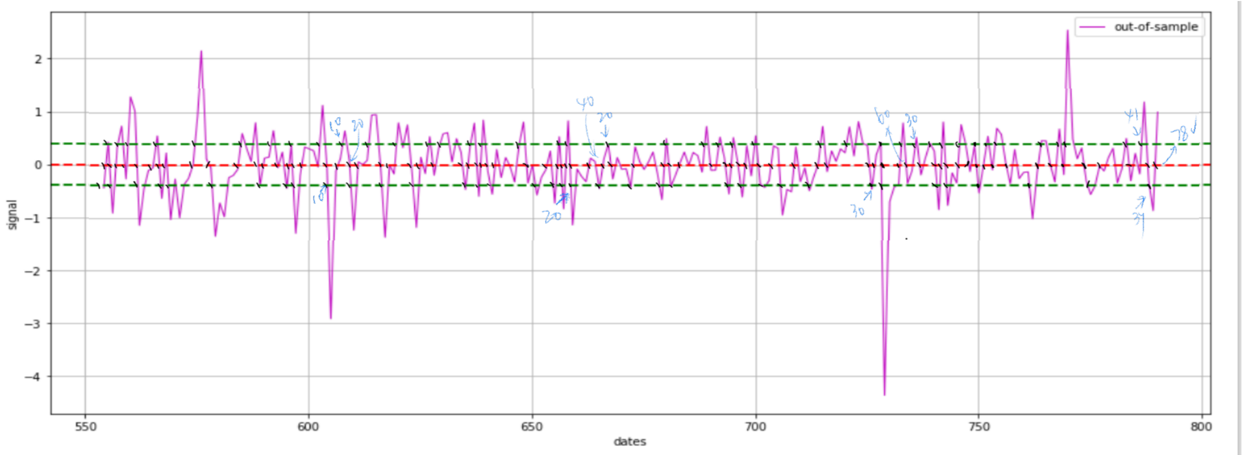


Figure 6: Out-of-sample test

4 Pair trading strategy

4.1 Introduction of Pairs Trading Strategy

Pairs Trading is a market neutral trading strategy enabling traders to profit from virtually any market conditions. This strategy monitors the close prices of two historically related securities. When the correlation between the two securities temporarily weakens, short the stock that performs well, and long the stock that performs poorly. The strategy holds that the difference between two stocks' prices is converging.

4.2 Match Co-integrated Stock

Thus, the key of pairs trading is finding stocks with highly correlated price movements(co-integrated stock), so that $S(t) = P_1(t) - w \times P_2(t)A$ will revert to zero.

4.2.1 Dataset and Preprocessing

Visa belongs to Credit Services Industry, so I listed 37 stocks from the same industry.[3] Due to the same main business, these stocks are correlated.
My sample data starts from 2021-01-01 to 2021-06-01, and out-of-sample data is from the rest of 2021. Since I only need to find one co-integrated stock, I remove all null values from data set.

4.2.2 Co-integrated Stock

Function **coint** is available to test co-integration, via constructing a p-values matrix. The lower p value is, the higher cointegration between two stocks. And I create a heat map to visualize the p-values matrix (figure 7).
The P-values between Visa and 'OCSL','MBNKP','NNI','SNFCA','AVG' are 0.00054, 0.000598, 0.00919, 0.012 and 0.042. So, Oaktree Specialty Lending Corporation (OCSL) is the most co-integrated stock with Visa in the first half of 2021.[4]

[3] **Website:** swingtradebot.com

[4] **Co-integration:** Co-integrated Stock is not fixed, since co-integration with visa changes over time. In the second half of year, the lowest p-values appears between visa and Nelnet (NNL)

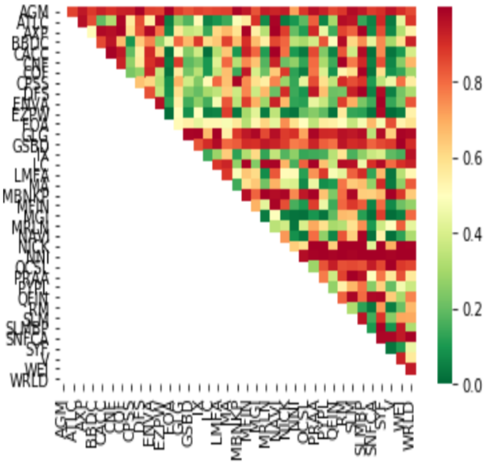
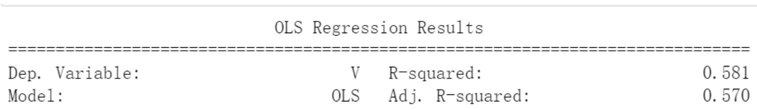


Figure 7: Heat Map of P-values Matrix

4.2.3 Evaluation of Linear Regression Model

The purpose of pairs trading is to construct $S(t) = P_1(t) - w \times P_2(t)A$. The plotting of OCSL-V is linear, and R-squared is 0.581, which proves linear relationship to a certain extent.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      V   R-squared:                       0.581
Model:                            OLS   Adj. R-squared:                  0.570
```

Figure 8: Pairs-trading OLS Linear Regression Result

4.3 Pairs Trading Strategy

When constructing Pairs Trading strategy, I set the threshold as the standard deviation of residual (0.283). Traders should long Visa and OCSL when signal hits the level (mean-std); short Visa and long the OCSL when signal hits the level (mean+std), as the figure 9 shows.

Traders will execute three shorts and two long. Each trade earns \$ 0.283, and the total revenue is \$ 1.415 per share.

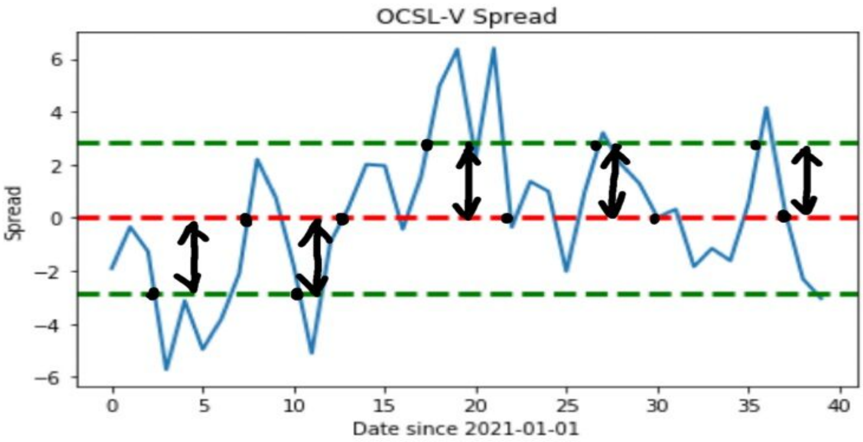


Figure 9: Pairs Trading Strategy

4.4 Out-of-sample test

When practicing the same pairs trading strategy towards out-of-sample data, traders can still earn some money. But the trategy is not that profitable, because most of the curve is not fluctuated within the threshold. Traders can only long visa (short OCSL) for one time, and short Visa (long OCSL) for two times. Each trade brings $ 0.283, so the total revenue is $ 0.85 per share.

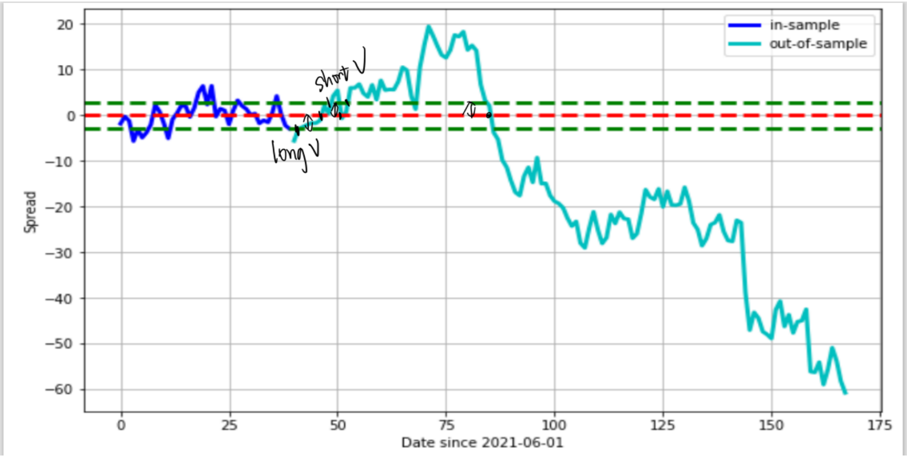


Figure 10: Pairs Trading Strategy

5 Conclusion

Since the time periods of sample-data or out-of-sample data of two strategies matches during 2021, I will compare them by adding up total profit of 2021.

During 2019-2020, dynamic trading strategy based on PCA method brings $ 3.94 per share. It brings $ 30.79 per share during 2021. As for Pairs Trading strategy, it provides $ 1.45 per share during 2021/01–2021/05, and provides $ 0.85 per share during 2021/06–2021/11.

Apparently, dynamic trading strategy enable traders to execute more frequent transaction and is more profitable.

Apart from that, Pairs Trading strategy is not convenient as well. Traders need to take great effort to search co-integrated stocks, and the co-integration or correlation between two stocks is not always significant. Thus, dynamic trading strategy is a better option.