ORIE 4741 Course Project Final Report
Application of Machine Learning Techniques in Statistical Arbitrage

Zicheng Men, Sheng Zhang, Shan He

December 4 2016

1 Introduction

The purpose of this project is to develop a trading strategy that tracks arbitrage opportunities in a stock index. The net present value (NPV) of a stock index is computed from the weighted average of the prices of selected stocks. In the market, people can trade the stock index using exchange traded fund (ETF) of the index. On the other hand, the actual trading price of the ETF is determined by the relationship between supply and demand of the market. Therefore, the discrepancy between its NPV and actual trading price creates arbitrage opportunities. It is commonly practiced to use constituents of the index to predict the index fluctuation.

We first employ linear regression to study the relationship between the price of the ETF and the portfolio of the index's constituents. In order to solve the overfitting problem we observe in the linear model, we propose Principal Component Analysis (PCA) and different regularizations as competing models. As a result, PCA gives better training and testing errors. Then we further conduct the Autoregressive Moving Average (ARMA) model on the residuals of the PCA model to model the difference between the ETF and the portfolio of constituents. The trading signal is generated by comparing the difference with a certain threshold value. Finally, we back-test our trading strategy and conclude that our strategy is profitable.

2 Data Selection

We choose to look into the S&P 100 index and its constituents which are 102 leading US stocks, including Apple, Microsoft, etc. We use the adjusted close prices of 101 stocks in S&P 100 as our feature space, and use adjusted close price of iShares S&P 100 ETF as our output space. We exclude one of the constituents because it has missing data from April to June. We consider the data of 102 days from April. 11, 2016 to Sep. 2, 2016. To avoid overfitting parameters, we use this 102-day period as our training set because we need at least 102 observations to train the 102 parameters (including an offset term). We download the data from Yahoo Finance using R.

3 Linear Regression

To begin with, we do a basic linear regression using least squares (LS) fitting. The training error on the 101 constituents over 102 training days is around 10^{-3}. Then we improve our model by adding offset and standardizing features to input data. The standardizing formula we use is $\frac{X_t-\mu}{\sigma}$, where μ and σ is the mean and standard deviation respectively.

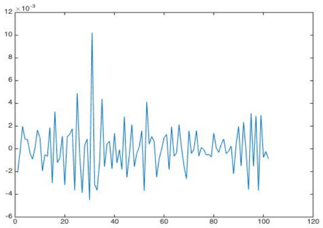

Figure 1: Training error of the basic linear model

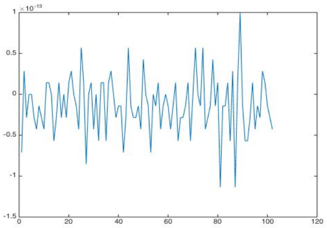

Figure 2: Training error of the adding offset model

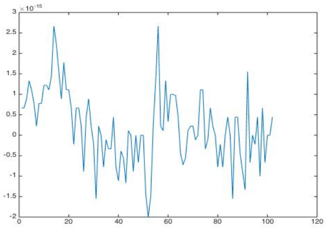

Figure 3: Training error of standardized model w/ offset

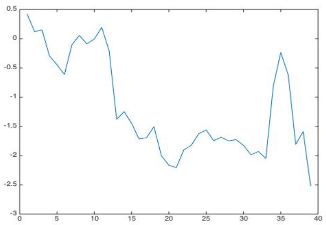

Figure 4: Testing error of the basic linear model

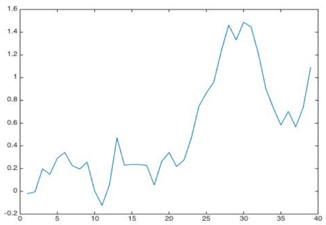

Figure 5: Tesing error of the adding offset model

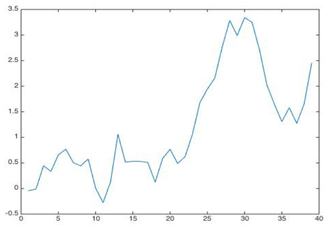

Figure 6: Testing error of standardized model w/ offset

From Figures 1 2 and 3, we observe that the training errors of three models are acceptable (at the levels of 10^{-3}, 10^{-13}, 10^{-15} respectively). However, when we test this linear model using 39 examples that are not in the training set, i.e. adjusted close prices of each stock from Sep. 5, 2016 to Oct. 27, 2016, the testing errors of the three models are far from satisfactory, as shown in Figures 4 5 and 6. Please note that we plot the error of each date so the x axis is the date.

This implies that we are facing an overfitting problem, even though we have used the smallest possible training set. There are too many parameters (dimensions) in the linear model, which gets us into this problem. To tackle this problem, we propose two competing approaches. One is to add different regularizers ($l1$, $l2$) to directly solve the overfitting issue. The other is to conduct the Principal Component Analysis(PCA) to reduce the feature dimensions. Both approaches are done on the basis of the feature-standardized linear model.

4 Solving Overfitting

4.1 Regularization

We know that regularization can effectively solve the overfitting problem. We add $l1$ and $l2$ regularizers to the original feature-srandardized linear model. So we need to solve the following optimization problems:

$$\begin{aligned} l1\ regularizer : minimize \quad & \sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{n} |w_i| \\ l2\ regularizer : minimize \quad & \sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{n} w_i^2 \end{aligned} \tag{4.1}$$

We applied proximal gradient method to quadratic loss function with $l1$ and $l2$ regularizers. At first, we picked 2000 as maximum number of iterations. We test the models on the testing set and the results are not very satisfying. As a result, we tried 20000 iterations to see if it is going to improve the outcome. As it turns out, 20000 iterations give better results and the testing errors of $l1$ and $l2$ are similar (Figure 7 and Figure 8).

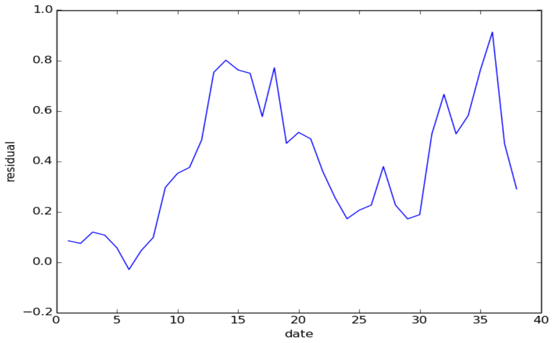

Figure 7: Testing error of linear model with l1 regularizer

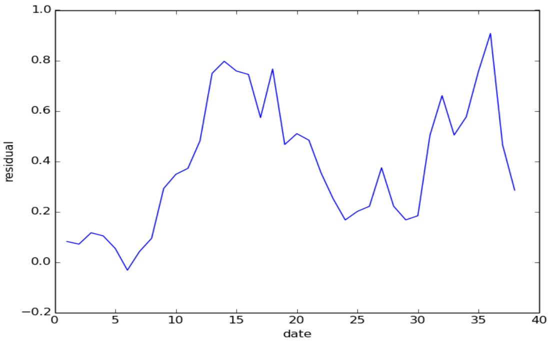

Figure 8: Testing error of linear model with l2 regularizer

4.2 Principal Component Analysis

The other way to solve this issue caused by the large input space is to reduce the feature dimensions, which can be done through the PCA. We use the previous 102 training days to estimate the correlation matrix of 101 features. Since PCA exports the principal components in the descending significance order, we compute the eigenvalues of the correlation matrix and observe that the first 20 eigenvalues reveal almost all information of the correlation matrix. In Figure 9, we plot the relative eigenvalues which is the absolute eigenvalue divided by the sum of all eigenvalues. The relative eigenvalues can be interpreted as the percentage of market information explained.

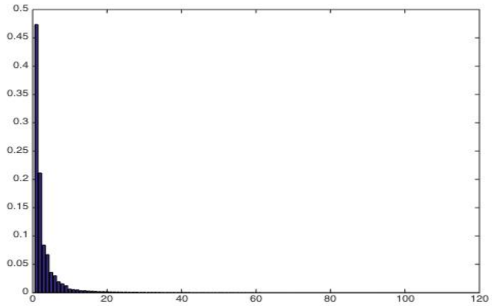

Figure 9: Training errors of first 14 PC

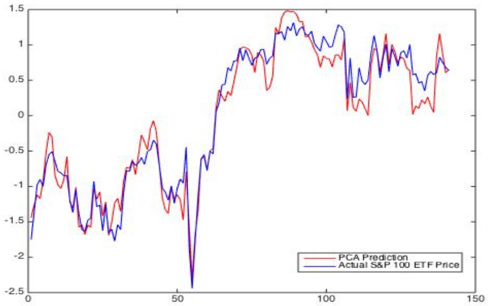

Figure 10: Fitted PCA vs ETF Price

By comparing the training errors of model with first one principal component (PC), first two PC, ..., and first 20 PC, we choose first 14 PC as the features of the new model since the first 14 has the least errors from both training set and testing set.

4.3 Approaches Comparison

By comparison, we conclude that the PCA approach achieves smaller training and testing error. In Figure 10, we compare the price evolution of the fitted PCA linear model and the actual iShares S&P 100 ETF price in both training and testing sets. In the following sections, we will use the residuals of the regression of the first 14 PC to generate trading signals.

5 ARMA Model

We consider the price of the constituent portfolio as the expected price of the index. The difference between the price of the ETF and its constituents portfolio is due to the idiosyncratic risk of the ETF itself. Since the residual terms measures the idiosyncratic risks of the ETF, we want to model the residual such that we can determine whether to trade or not by studying the residual. According to empirical studies, the idiosyncratic risk tends to converge to its long term mean, the overall industry level. Therefore, we consider the residual term as a stationary stochastic process. To model a stationary stochastic process, one suitable candidate is the Autoregressive - Moving Average (ARMA) model.

The ARMA model is the combination of Autoregressive (AR) model and Moving Average (MA) model. The AR part involes regressing the variable on its own lagged values (past values), while the MA part models the error term as a linear combination of past error terms. According to the definition of ARMA(p,q) model (p for the order of AR part, q for the order of MA part), the model can be written as:

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i a_{t-i} + a_t, \quad a_i \sim White\ Noise(0, \sigma_a^2) \tag{5.1}$$

In our case, X_t is the residual values. We use the residuals generated from the PCA model in the training set as our input space. [1].

5.1 Model Construction

In this section, we are going to construct the ARMA model. According to the definition of ARMA model, in order for the model to work, we need the time series data to be weakly stationary. Weakly stationary property requires the first momenet (mean) of the input to not vary with respect to time:

$$\mathbb{E}(X_t) = m_X(t) = m_X(t + \tau), \quad \forall \tau \in \mathbb{R} \tag{5.2}$$

To check the weak stationarity, the Augmented Dickey-Fuller test, also known as unit root test, was conducted. The result shows that the P-value is less than 0.05. Therefore, we would reject the null hypothesis and conclude that the time series data is stationary.

To determine the order, we use the Akaike Information Creteria (AIC).

$$AIC = 2k - 2\ln(L) \tag{5.3}$$

Where k is the number of parameters and L is the maximized value of the likelihood function.

ARMA Order	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
$q = 1$	-74.52001	-75.40049	-74.95146	-73.35212	-72.51332	-70.94867
$q = 2$	-73.88572	-74.77141	-73.72606	-71.87368	-74.46360	-69.00147
$q = 3$	-73.00846	-73.51266	-71.85228	-69.84067	-68.57158	-70.77314
$q = 4$	-71.80729	-73.16802	-71.25763	-67.85268	-72.86408	-70.73413
$q = 5$	-70.87616	-70.00024	-68.00054	-70.77085	-70.36064	-69.82982
$q = 6$	-70.75341	-74.07904	-72.13562	-68.86078	-67.54460	-69.25445

Table 1: AIC Value Table

[1] For more information on ARMA model, please visit `https://en.wikipedia.org/wiki/Autoregressive-moving-average_model`

As we can see from the above table, AIC achieves the lowest value at ARMA(2,1). Therefore, we will proceed with ARMA(2,1). With parameters estimated using Maximum Likelihood method, we can write down the ARMA(2,1) model explicitly as the following:

$$X_t = 0.0011 + 0.9193X_{t-1} - 0.3408X_{t-2} + a_t - 0.2937a_{t-1} \tag{5.4}$$

5.2 Model Adequacy

To check the model's adequacy, we perform the Box-Ljung test on the residuals to check if there is serial correlation and Box-Ljung test on the squared residuals to check if there is ARCH effect (Autoregressive Conditional Heteroskedasticity effect). ARCH effect exists when the variance of the current error term is actually a function of past error terms, while the ARMA model assumes the variance of error terms to be constant. The test results are as the following:

Box-Ljung Test
X-squared = 14.3476, df = 12, p-value = 0.2791 Conclusion: We fail to reject the null and conclude that no serrial correlation

Table 2: Box-Ljung Test for serrial correlation of ARMA(2,1)

Box-Ljung Test
X-squared = 16.7294, df = 12, p-value = 0.1601 Conclusion: We fail to reject the null and conclude that no ARCH effect exists

Table 3: Box-Ljung Test for ARCH effect of ARMA(2,1)

Therefore, we can conclude that our model is adequate.

6 Trading Signal Generation

Since the residual is the difference between the actual iShares S&P 100 ETF price and our PCA model's prediction, a positive (negative) residual suggests that the ETF is overpriced (underpriced) therefore we should buy in (sell out) the portfolio of S&P 100 constituents if the magnitude of the difference is big enough. Ideally, the difference should be bigger than its empirical average level and big enough to cover the transaction fee. Since the residuals have been modeled by our ARMA model, we define the trading signal as:

$$TS = \frac{X_t - m_{equilibrium}}{\sigma_{equilibrium}} \tag{6.1}$$

where $m_{equilibrium}$ and $\sigma_{equilibrium}$ are the theoretical mean and standard deviation of the residual in equilibrium according to the ARMA model.

In equilibrium, we have

$$\begin{aligned} X_t &= 0.0011 + 0.9193X_{t-1} - 0.3408X_{t-2} + a_t - 0.2937a_{t-1} \\ X_t &= X_{t-1} = X_{t-2} \end{aligned} \tag{6.2}$$

Therefore, by taking expectation and variance on both sides of the ARMA model, we can calculate the equilibrium mean, $m_{equilibrium}$, and standard deviation, $\sigma_{equilibrium}$. Note that the white noise $a_i \in N(0, \sigma_a^2)$, $\forall\ t > 0$. According to the ARMA model, $\sigma_a = 0.02559$

$$\begin{aligned} &\mathbb{E}[X_t] = 0.0011 + 0.9193\mathbb{E}[X_{t-1}] - 0.3408\mathbb{E}[X_{t-2}] + \mathbb{E}[a_t] - 0.2937\mathbb{E}[a_{t-1}] \\ &\mathbb{E}[X_t] = \mathbb{E}[X_{t-1}] = \mathbb{E}[X_{t-2}] = m_{equilibrium} \\ &\implies m_{equilibrium} \approx 0.0032211 \end{aligned}$$

$$\begin{aligned} \mathrm{Var}[X_t] &= \mathrm{Var}[0.0011 + 0.9193X_{t-1} - 0.3408X_{t-2} + a_t - 0.2937a_{t-1}] \\ &= 0.8451\,\mathrm{Var}[X_{t-1}] + 0.1161\,\mathrm{Var}[X_{t-2}] - 0.6266Cov(X_{t-1}, X_{t-2}) + \mathrm{Var}[a_t] + 0.0863\,\mathrm{Var}[a_{t-1}] \\ \implies \sigma_{equilibrium} &\approx 0.2573023 \end{aligned} \tag{6.3}$$

Therefore, the trading signal is $TS = \frac{X_t - 0.0032211}{0.2573023}$.

The trading signal measures the difference of the portfolio of S&P 100 constituents and the iShares S&P 100 ETF price per unit standard deviation. Recall that we need the trading signal to be strong enough to cover the transaction fee. So a 'buy' signal is generated only if TS is strictly bigger than a certain threshold value and a 'sell' signal is generated only if TS is strictly smaller than a certain threshold value. We believe that the threshold value should be a function of the initial funding and transaction fee. Moreover, different investors can have different threshold value according to their degree of risk aversion. Therefore, we do not specify the threshold value here, but we will provide an example in the back-testing section.

7 Back Testing

Because different investors have different trading behaviors, our approach of back testing can by no means represent all the possible senarios. The assumptions we make are as the follows:

- The initial funding is $100,000.
- The commission fee per trade is $7.95 (according to *Fidelity Stock Trading*). Since each 'buy'(or 'sell') involves buying all 101 constituent stocks with weights equalling the actual weight in S&P 100 index and selling the S&P 100 ETF, one arbitrage transaction is 102 trades in total. So the commission per one transaction is $811.
- There is no other transaction fee excluding commission.

- For each trade, we buy in stocks with total value equalling \$100,000 and sell out the same amount. So, in fact, we use the proceeds from the short selling to buy. Our initial funding only serves as reserve. We do not use the initial funding to trade.

- We can only engage in trading if the trade closes our stock positions at hand after the first transaction. In other words, if we buy the constituents portfolio and sell ETF in last transaction, we can only sell constituents portfolio and buy ETF in this transaction. It is to guarantee that we can always cover the loss from short selling using our reserve in case our trading strategy fails.

- The trading horizon is 39 days, using our test set data.

- The threshold value is calculated based on the principle that the predicted profit from one transaction can at least cover the commission.In this back test, the threshold values are, 'buy': 0.7082 and 'sell': -0.7082.

According to our back testing result, our statistical arbitrage trading strategy yields an 3.87% return in the 39 trading horizon. Assuming 252 trading days in a year, the annualized rate of return is 24.99%.

8 Conclusion

According to the back testing result, our trading strategy is actually profitable. We are confident in our result because we use a very conservative approach in back testing. We can achieve even bigger return rate if we use smaller threshold value (institutional trading can often have lower commission than individuals) and higher leverage. However, we admit that there are flaws in our strategy. For example, we do not test on extreme periods, such 2008 financial crisis, and we do not know if our strategy can work on other stock index. Therefore, more work on these aspects can be done in the future.