Statistical Arbitrage Using Pairs Trading With Support Vector Machine Learning

by

Gopal Rao Madhavaram

A research project submitted in partial fulfillment of

the requirements for the degree of Master of Finance

Saint Mary's University

Written for MFIN 6692.0 under the direction of Dr. J. Colin Dodds

Approved: Dr. J. Colin Dodds

Faculty Advisor

Approved: Dr. Francis Boabang

MFIN Director

Date: August 26, 2013

Acknowledgements

I would like to express my special thanks of gratitude to my supervisor Dr. J. Colin Dodds as well as MFIN Director Dr. Francis Boabang who gave me the golden opportunity to do this wonderful project on the topic “Statistical Arbitrage Using Pairs Trading With Support Vector Machine Learning”, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me in finalizing this project within the limited time frame.

Abstract

Statistical Arbitrage Using Pairs Trading With

Support Vector Machine Learning

by

Gopal Rao Madhavaram

The purpose of this study is to analyze the performance of dynamic PCA (Principal Component Analysis) Statistical Arbitrage, and to validate the results with the help of a novel Machine Learning approach known as Support Vector Machines using the "Pairs trading" strategy. The paper starts by explaining the fundamental concepts behind our analysis e.g. Linear Regression, Auto-Regressive processes and Orstein Uhlenback modeling of residuals. Research focus will be on two things: how the principal components are obtained and how the portfolio of systematic risk factors is formed.

Stock data of 20 stocks from the XLF financial sector is chosen for the principal components analysis. The data includes each stock's daily opening price, high, low, adjusted close price and daily volume from the year 1998 to 2012. There are total of 69,920 observations.

The paper concludes by demonstrating the scenario when SVM gave better results compared to the basic Mean-reversion strategy and future enhancements possible with this mixed approach.

Table of Contents

CHAPTER 1: INTRODUCTION 1
1.1 Purpose of Study 1
1.2 Background 1
1.2 Need for study 3
1.3 Statement of purpose 4
Chapter 2: Literature Review 6
Chapter 3: Methodology 9
3.1 Introduction to Research Design 9
3.2 Sampling Design & Data Collection 9
3.3 Statistical Arbitrage 10
3.3.1 Linear Regression: 10
3.3.2 O-U (Ornstein-Uhlenback) Process: 13
3.3.3 Trading Strategy: 15
3.3.4 Principal Component Analysis: 17
3.4 SVM (Support Vector Machines) 21
3.4.1 Empirical Risk Minimization: 22
3.4.2 Regression for Classification: 24
3.4.3 Structural Risk Minimization: 25
3.4.3 VC (Vapnik–Chervonenkis) Dimension: 27
3.4.4 Optimization Steps: 28
3.4.5 Non-Linear classification using SVM: 30
Chapter 4: Results 31
4.1 Results for PCA Strategy: 31
4.2 Performance with SVM Validation: 33
Chapter 5: Conclusions & Feasible Future Enhancements 36
References 38
Appendix A: Code for SVM Learning 41

CHAPTER 1: INTRODUCTION

1.1 Purpose of Study

The goal of this project is to analyze the performance of dynamic PCA (Principal Component Analysis) Statistical Arbitrage, and to validate the results with the help of a novel Machine Learning approach known as Support Vector Machines using the "**Pairs trading"** strategy.

The global stock market is characterized with great uncertainty and risks. However, many investors make profits using the available information and by implementing trading strategies. Many investors use technical analysis to buy stocks of particular companies, others use strategies based on market behaviour. "Pairs trading" is one of those strategies used to detect arbitrage opportunities in the stock market. Pairs' trading is the ancestor of Statistical Arbitrage. The idea behind pairs trading is that, if stocks P and Q belong to the same industry or have similar characteristics, then one expects the returns of the two stocks to track each other after controlling for beta.

1.2 Background

Pairs' trading was originally developed by a group of computer scientists, physicists and mathematicians employed by Morgan Stanley & Co. in the 1980s. The team comprised of/ included computer scientists, Gerry Bamberger and David Shaw,

and quant trader Nunzio Tartaglia, who studied the arbitrage opportunities. This led to the development of automated trading program using advanced statistical modeling to exploit the market uncertainties.

The result of their research is the development of a quantitative strategy to identify pairs of securities that are highly correlated and exhibit similar historical price movements. The method called black box proved to be successful in 1987 where the group made a $50 million profit for Morgan Stanley. However, in the next two years the method gave poor results which led to the termination of the group in 1989. This method was a revolution at that time and many models or strategies were developed leading to extensive use of quantitative methods and time series data. Tartaglia's own explanation for pairs trading is psychological. He claims, that "…Human beings don't like to trade against human nature, which wants to buy stocks after they go up not down.[1]" (Hansell, 1989)

Although the group was disbanded in 1989, "pairs trading" has since become very popular among individual and institutional investors as a "market-neutral" strategy.

Recently, Vapnik and his colleagues have developed a novel neural network algorithm called support vector machine (SVM). Structural risk minimization is implemented by the SVM. Empirical risk minimization principles were used by many traditional neural network models. The traditional models seek to reduce the mis-classification error from proper solution of training data but SVM minimizes the upper bound of generalization error. The SVM gives the global optimum solution while the traditional neural network

models falls into the local optimum solution. Thus, there is no chance of over-fitting with SVM.

1.2 Need for study

The main goal of any investor is to earn profit from their investment without losing any initial invested capital. Earning profits has become very difficult due to the uncertainties and risks involved in stock market. So the implementation of certain trading strategies has become very useful in exploiting the market by using statistical arbitrage.

The term statistical arbitrage includes various strategies and investment methods. The common features in them are: (i) trading signals are systematic and not driven by fundamentals, (ii) the trading book is market-neutral, i.e., it has zero beta with the market, and (iii) the method of generating excess returns is statistical. The goal is to make many investment bets with positive expected returns by taking advantage of diversified portfolios across stocks and to produce a less volatile investment strategy which is highly uncorrelated with the market.

So this research paper uses pairs trading to analyze the performance of the dynamic PCA statistical arbitrage using the support vector machine language. The exploitation of arbitrage was extensively used by many institutional investors and hedge funds to make lots of profits.

1.3 Statement of purpose

The goal of this project is to analyze the performance of dynamic PCA Statistical Arbitrage, and validate the results with a novel Machine Learning approach known as Support Vector Machines.

The idea behind pairs trading is that if the stocks P and Q belong to the same industry or have similar characteristics then one expects the returns of the two stocks to track each other after controlling for beta.

In mathematical form:

$$\frac{dP_t}{P_t} = \alpha dt + \beta \frac{dQ_t}{Q_t} + dX_t \quad \text{.....(1.1)}$$

where:

P and Q are stocks from the same industry.

X_t is referred to as the co integration residual.

β is dollar amount of stock Q to be shorted.

In many cases of interest, the drift α is small compared to the fluctuations of X_t and can therefore be neglected.

The paper expects that if this residual deviates to an extreme, it will revert to the equilibrium soon, according to the Mean-reversion principle. This model suggests an investment strategy where we go long 1 dollar of stock P and short β dollars of stock Q if X(t) is at negative extreme end and conversely, go short P and long Q if X(t) is at positive

extreme. The research analyzes the nature of residuals of mean-reversion to form a trading rule around the residual process. The way this paper generates the residual series is by examining the co-integration characteristics of a set of stocks in the financial sector ETF-XLF with a set of systematic risk factors associated with the market. Note that here the emphasis is on the residual that remains after the decomposition is done. Our approach of generating systematic factors is using PCA (Principal component analysis).

Chapter 2: Literature Review

Several studies have been conducted on detecting arbitrage in the stock market and how investors use some strategies to profit from the arbitrage opportunities. Most of the strategies are based on quantitative techniques. In their research paper Gatev, Goetzmann and Rouwenhorst stated that ……

"*Wall Street has long been interested in quantitative methods of speculation. One popular short-term speculation strategy is known as "pairs trading." The strategy has at least a twenty year history on Wall Street and is among the proprietary "statistical arbitrage" tools currently used by hedge funds as well as investment banks." (Gatev, Goetzmann, & Rouwenhorst, 2006).*

They tested a Wall Street investment strategy, "pairs trading," with daily data over 1962-2002. With minimum distance between normalized historical prices, stocks are matched into pairs. An average annualized excess return of up to 11 percent for self-financing portfolios of pairs is yielded by using a simple trading rule. The profits typically exceeded conservative transaction cost estimates. Results from the Bootstrap suggest that the "pairs" effect differs from previously-documented reversal profits. The excess return indicates that pairs trading profits from temporary mis-pricing of close substitutes in the same industry. As opposed to conventional risk measures, they have linked the profitability to the presence of a common factor in the returns.

In his paper Kyoung Jae Kim stated that ….."*Support vector machines (SVMs) are promising methods for the prediction of financial time-series because they use a risk function consisting of the empirical error and a regularized term which is derived from the structural risk minimization principle. This study applies SVM to predicting the stock price index. In addition, this study examines the feasibility of applying SVM in financial forecasting by comparing it with back-propagation neural networks and case-*

based reasoning. The experimental results show that SVM provides a promising alternative to stock market prediction." (Kyoung-jae, 2003).

In a research conducted by Avellaneda and Lee (2008) the authors studied model-driven statistical arbitrage strategies in U.S. equities. They agreed that …..

"Trading *signals are generated in two ways: using Principal Component Analysis and using sector ETFs". (Avellaneda & Lee, 2008).*

In both cases, they considered the residuals, or idiosyncratic components of stock returns, and modeled them as a mean-reverting process, which leads naturally to "contrarian" trading signals. The main contribution of their paper is the back-testing and comparison of market-neutral PCA- and ETF- based strategies over the broad universe of U.S. equities.

Back-testing shows that, after accounting for transaction costs, PCA-based strategies have an average annual Sharpe ratio of 1.44 over the period 1997 to 2007, with a much stronger performances prior to 2003: during 2003-2007, the average Sharpe ratio of PCA-based strategies was only 0.9. On the other hand, strategies based on ETFs achieved a Sharpe ratio of 1.1 from 1997 to 2007, but experienced a similar degradation of performance after 2002. We introduce a method to take into account daily trading volume information in the signals (using "trading time" as opposed to calendar time), and observe significant improvements in performance in the case of ETF-based signals.

ETF strategies which use volume information and achieve a Sharpe ratio of 1.51 from 2003 to 2007. The paper also relates the performance of mean-reversion statistical

arbitrage strategies with the stock market cycle. In particular, they studied in some detail the performance of the strategies during the liquidity crisis of the summer of 2007.

The goal of this paper is to analyze the performance of dynamic PCA Statistical Arbitrage, and validate the results with a novel Machine Learning approach known as Support Vector Machines. The paper expects that if this residual deviates to an extreme, it will revert to the equilibrium soon, according to the Mean-reversion principle.

Chapter 3: Methodology

3.1 Introduction to Research Design

The paper starts by explaining the fundamental concepts behind our analysis e.g. Linear Regression, Auto-Regressive processes and Orstein Uhlenback modeling of residuals. Research focus will be on two things: how the principal components are obtained and how the portfolio of systematic risk factors is formed. The paper goes on to discuss the performance of this modeling and the results of applying the Support Vector Machines on top of our basic trading strategy using PCA. Support Vector Machine is a momentum-based approach as opposed to Mean-reversion. It is hoped that this research will make an optimal trading decision out of the two models.

The paper concludes by demonstrating the scenario when SVM gave better results compared to the basic Mean-reversion strategy and future enhancements possible with this mixed approach.

3.2 Sampling Design & Data Collection

For this research paper, Stock data of 20 stocks from the XLF financial sector is chosen for the principal components analysis:

ACE, AFL, AIG, AMT, AXP, BAC, BK, BLK, BRK.B, C, COF, GS, JPM, MET, PNC, PRU, SPG, TRV, USB, WFC.

The data includes each stock's daily opening price, high, low, adjusted close price and daily volume from the year 1998 to 2012. There are total of 69,920 observations. Once these stocks are selected, statistical arbitrage is performed on the first 10 stocks with systematic returns as the projection of original 20 stocks' returns onto the Eigen vectors and dynamically varying the Principal Components depending on the required amount of variance.

Further sections will discuss the results of the trading with the parameters mentioned in the previous sections and the application of Support Vector Machines over this basic trading system.

3.3 Statistical Arbitrage

3.3.1 Linear Regression:

The study decomposes stock returns into systematic and idiosyncratic components and establishes trading rules on residuals which is the paradigm of pairs-trading

$$R_i = \sum_{j=1}^{m} \beta_{ij} F_j + \tilde{R}_i \quad \text{.....(3.1)}$$

where:

R is the returns of stock.

F is the returns of systematic components.

It is important to note that if Q_i is the amount of money invested in stock i, we have,

$$\begin{aligned}\sum_{i=1}^{N} Q_i R_i &= \sum_{i=1}^{N} Q_i \left[\sum_{j=1}^{m} \beta_{ij} F_j\right] + \sum_{i=1}^{N} Q_i \tilde{R}_i \\ &= \sum_{j=1}^{m} \left[\sum_{i=1}^{N} \beta_{ij} Q_i\right] F_j + \sum_{i=1}^{N} Q_i \tilde{R}_i \end{aligned} \quad \text{…….. (3.2)}$$

Hence for the portfolio of stocks to be market-neutral, we must have

$$\sum_{i=1}^{N} \beta_{ij} Q_i = 0$$

Assuming beta neutrality, the task will be to decompose the stock returns in systematic factors. This is simply achieved by Simple Linear Regression. SLR gives models the relationship between a response variable *Y* and a predictor variable *X*.

$$y = \alpha + \beta x \quad \text{……...(3.3)}$$

It does this in such a way that the squared errors of residuals are minimized i.e.

$$\text{Find } \min_{\alpha,\beta} Q(\alpha,\beta), \text{ where } Q(\alpha,\beta) = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

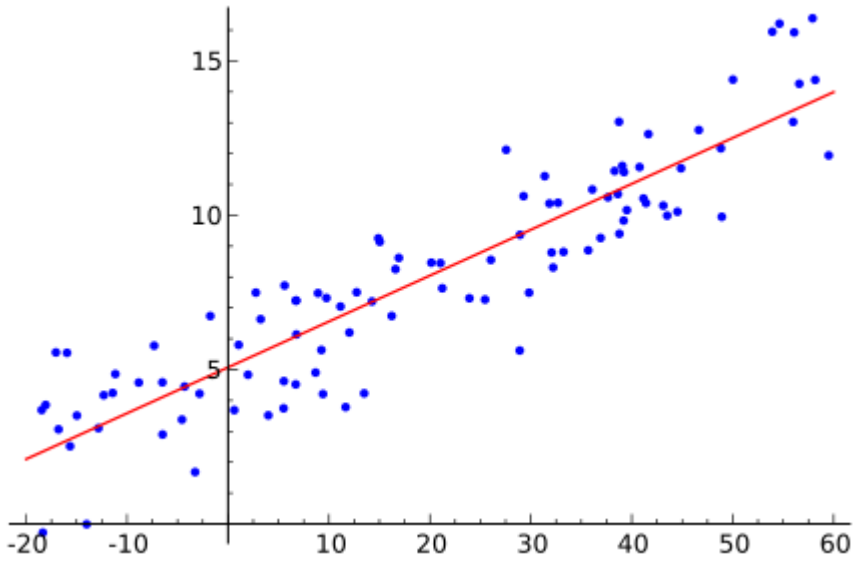
We can easily solve this problem by using elementary calculus and we obtain the optimal value of beta as:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{j=1}^{n} y_j}{\sum_{i=1}^{n}(x_i^2) - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2}$$

…….. (3.4)

After we get these, residuals are obtained by subtracting out the outcomes and the predicted values. In the current context these residuals correspond to the idiosyncratic components of the model described above for stock returns.

Figure 3.1:



The above plot in Figure 3.1 is a schematic representation of how the regression line typically looks like in a single variable case, accordingly a hyper-plane in a multivariate case.

3.3.2 O-U (Ornstein-Uhlenback) Process:

The important factors in research analysis are the residuals. The next task will be to model the residuals. By looking at the mean-reverting nature of the residuals, our obvious choice is the Ornstein-Uhlenback model. Roughly speaking, the OU process describes the velocity of a massive Brownian particle under the influence of friction. It bears both the properties of our interest – stationarity and mean-reverting. So, the residuals for stock i can be modeled as follows:

$$dX_i(t) = \kappa_i \left(m_i - X_i(t)\right) dt + \sigma_i dW_i(t), \quad \kappa_i > 0 \qquad \text{..... (3.5)}$$

where:

X is the residual.

W is the weiner process.

σ is the standard deviation.

This is a stochastic differential equation and the parameters are specific to each stock. They are assumed to vary slowly in relation to the Brownian motion increments dWi(t), in the time-window of interest. This paper estimates the statistics for the residual process on a window of 60 days, assuming that the parameters are constant over the window. This hypothesis is tested for each stock in the universe, by goodness-of-fit of the model and, in particular, by analyzing the speed of mean-reversion. If assumed momentarily that the parameters of the model are constant, the equation can be written as,

$$X_i(t_0 + \Delta t) = e^{-\kappa_i \Delta t} X_i(t_0) + \left(1 - e^{-\kappa_i \Delta t}\right) m_i + \sigma_i \int_{t_0}^{t_0+\Delta t} e^{-\kappa_i (t_0+\Delta t-s)} dW_i(s)$$

…….…….. (3.6)

The parameter k can be thought of as the speed of mean-reversion. If k>>1 the stock reverts quickly to its mean and the research is interested in the stocks with fast mean-reversion.

It's important to observe that the discrete version of above equation corresponds to the Auto-regressive process. The study is interested in the AR process of order 1 which is defined as,

$$X_t = c + \varphi X_{t-1} + \varepsilon_t$$ …… (3.7)

where:

φ's are the parameters of the model.

Ɛ corresponds to white noise.

Now we can construct the discrete version of the OU process by defining,

$$X_k = \sum_{j=1}^{k} \epsilon_j \quad k = 1, 2, ..., 60$$ ……(3.8)

Note here that analysis will be based on the past 60 days where the study assumes that the parameters are constant.

One more thing to observe here is that the Auto-regressive process can simply be estimated by the Simple Linear Regression analysis. Therefore, the OU process boils down to the following regression model:

$$X_{n+1} = a + bX_n + \zeta_{n+1}, \quad n = 1, ..., 59 \qquad \text{..... (3.9)}$$

From the solution to the stochastic differential equation described above, we have,

$$\begin{aligned} a &= m\left(1 - e^{-\kappa\,\Delta t}\right) \\ b &= e^{-\kappa\,\Delta t} \\ \text{Variance}(\zeta) &= \sigma^2 \frac{1 - e^{-2\kappa\,\Delta t}}{2\kappa} \end{aligned}$$

$$\begin{aligned} \kappa &= -log(b) * 252 \\ m &= \frac{a}{1-b} \\ \sigma &= \sqrt{\frac{\text{Variance}(\zeta) \cdot 2\kappa}{1 - b^2}} \\ \sigma_{eq} &= \sqrt{\frac{\text{Variance}(\zeta)}{1 - b^2}} \end{aligned}$$

Here, note that X_{60}=0 which is an artifact of regression. Fast mean-reversion (compared to the 60-day estimation window) requires that k > 252/30 and based on this test we can reject/accept the stock.

3.3.3 Trading Strategy:

Once the regression analysis is done on the residuals, the next step is to form the trading rule. This can be done using the s-score which is defined as follows:

$$s = \frac{X(t) - m}{\sigma_{eq}} \qquad \text{..... (3.10)}$$

And since X_{60} is 0,

$$s = \frac{-m}{\sigma_{eq}} = \frac{-a \cdot \sqrt{1-b^2}}{(1-b) \cdot \sqrt{\text{Variance}(\zeta)}} \quad \text{..... (3.11)}$$

Here the basic trading signal based on mean-reversion is,

$$\begin{aligned} \text{buy to open if} \quad & s_i < -\overline{s}_{bo} \\ \text{sell to open if} \quad & s_i > +\overline{s}_{so} \end{aligned}$$

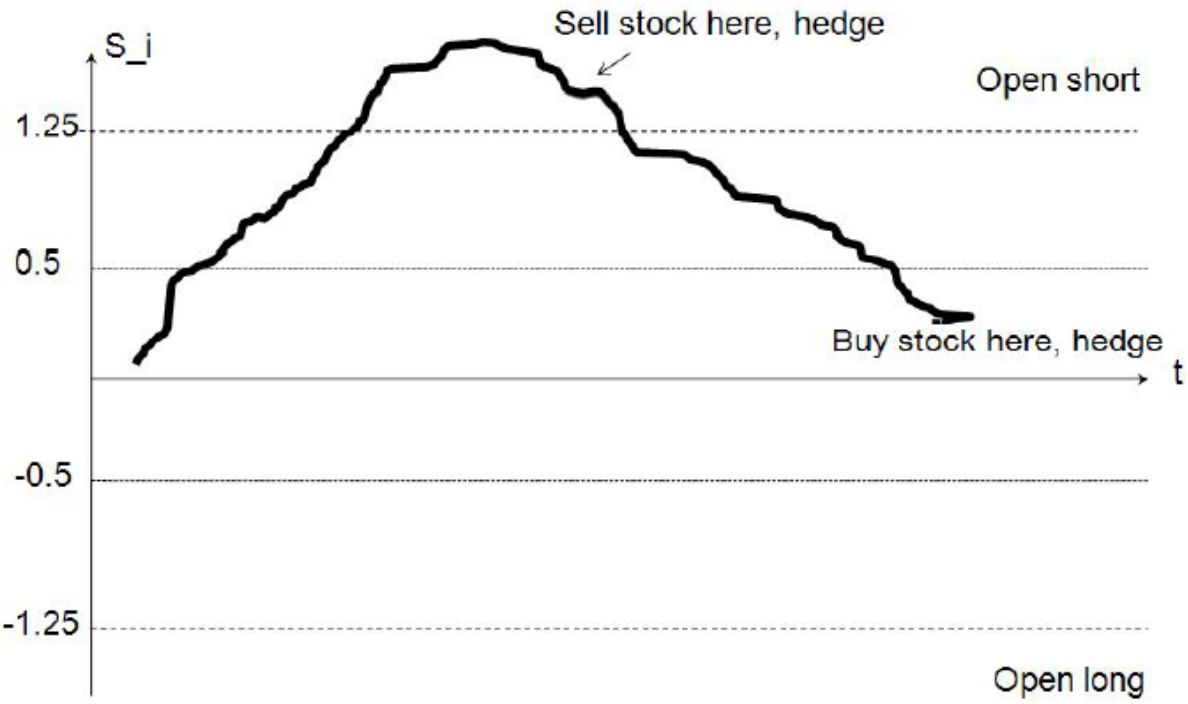
$$\begin{aligned} \text{close short position if} \quad & s_i < +\overline{s}_{bc} \\ \text{close long position} \quad & s_i > -\overline{s}_{sc} \end{aligned}$$

Buy to open means buying one dollar of the corresponding stock and selling beta dollars of systematic factors and similarly for other trading decisions. From the analysis, good values of these 'thresholds' are found to be,

$$\overline{s}_{bo} = \overline{s}_{so} = 1.25$$
$$\overline{s}_{bc} = 0.75 \quad \text{and} \quad \overline{s}_{sc} = 0.50$$

The below Figure 3.2 shows the evolution of s-score and the trading decisions based on this analysis:

Figure 3.2:



3.3.4 Principal Component Analysis:

As described in the beginning, Principal Component Analysis extracts the systematic risk factors. This approach uses historical share-price data on a cross-section of an arbitrary selection of N stocks going back to M days in history.

PCA is the orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. In the context, initial co-ordinate system corresponds to normalized returns of any arbitrary selection of stocks defined by,

$$Y_{ik} = \frac{R_{ik} - \overline{R}_i}{\overline{\sigma}_i} \quad \text{..... (3.12)}$$

$$\overline{R}_i = \frac{1}{M}\sum_{k=1}^{M} R_{ik} \quad \text{..... (3.13)}$$

$$\overline{\sigma}_i^2 = \frac{1}{M-1}\sum_{k=1}^{M}(R_{ik} - \overline{R}_i)^2 \quad \text{..... (3.14)}$$

Where, R represents the returns of various stocks.

$$R_{ik} = \frac{S_{i(t_0-(k-1)\Delta t)} - S_{i(t_0-k\Delta t)}}{S_{i(t_0-k\Delta t)}},\ k = 1, ..., M,\ i = 1, ..., N$$

.. (3.15)

Empirical correlation matrix of the data is given by:

$$\rho_{ij} = \frac{1}{M-1}\sum_{k=1}^{M} Y_{ik}Y_{jk} \quad \text{..... (3.16)}$$

Once the correlation matrix is in place, the way Principal Component Analysis works is by performing the Eigen Value Decomposition (EVD) of this matrix. Eigen values lambda of a matrix A are defined by the matrix equation,

$$\mathbf{Av} = \lambda\mathbf{v}$$

And the Eigen values are found by the equation,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad \text{..... (3.17)}$$

EVD of matrix A is therefore,

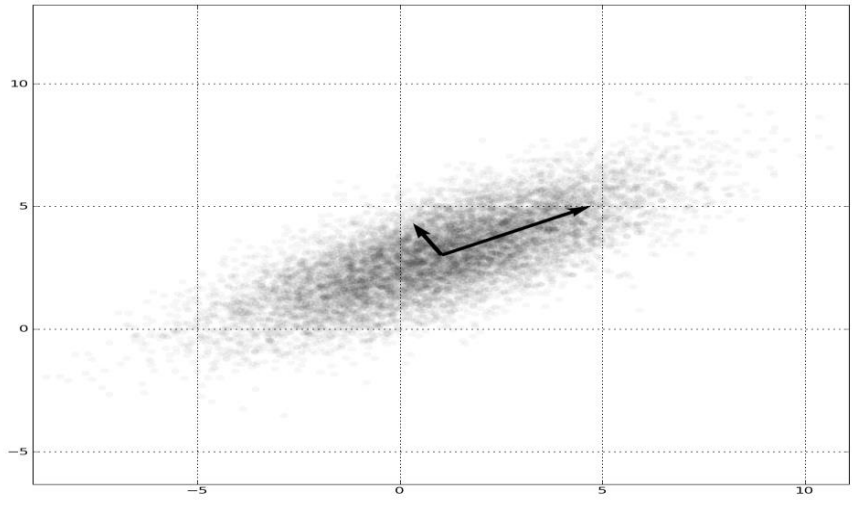
$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \quad \text{..... (3.18)}$$

where:

Q contains the Eigen vectors and the diagonal matrix Lambda contains the Eigen values.

A schematic diagram of Principal Components after EVD is shown below in Figure 3.3:

Figure 3.3:



(The thick black lines correspond to the Eigen vectors or the principal components, and the dots to actual data)

After the EVD of the return series of selected stocks, we obtain new return series' by projecting the original return series' onto the Eigen vectors. Every projection onto an Eigen vector will result in one new return series which corresponds to the systematic returns of our analysis.

$$F_{jk} = \sum_{i=1}^{N} \frac{v_i^{(j)}}{\overline{\sigma}_i} R_{ik} \quad \text{..... (3.19)}$$

where:

$$v^{(j)} = \left(v_1^{(j)}, ..., v_N^{(j)}\right) \ , \quad j = 1, ..., N$$

are the Eigen vectors.

The number of Principal Components (found by the Eigen vectors selected that explain a pre-determined amount of variance) is equal to the number of systematic factors chosen for our analysis (which varies across time).

These return series' together constitute what can be called an "Eigen portfolio" and are completely un-correlated (because of orthogonal decomposition). This results in more stable estimates of betas than with any other procedure.

The paper may choose only a specific number of factors in the Eigen portfolio depending on the desired level of variability to be explained, so we can control the parameter – "percentage of variance explained" to balance the trade-off between complexity and performance of our model.

The study is based on a dynamic selection of significant factors daily, taking the return series' of past 1 year data, that explain a fixed amount of variance - 70% turned out to be the optimal parameter in this case.

3.4 SVM (Support Vector Machines)

As it's briefly described in the introduction, the second part of the trading strategy is to validate the mean reversion point of view on a possible trade with the next day stock direction forecast. The forecast is predicted using three technical indicators, namely:

ROC (rate of change = close (t)-close (t-5)/close (t-5))

Stochastic Oscillator %K = close-LL (5)/ HH (5)-LL (5)

Close Value Location= close-low-(high-close)/high –low

where:

HH (5) = highest high achieved in past 5 days.

LL (5) = lowest low for the past 5 days.

This paper has chosen these indicators because the study is concerned with the market direction for the immediate next day, hence information reflecting the difference in today's price with tomorrow's price is sought. For this reason we are providing indicators which are capable of reflecting that price shift in series, rather than giving the stock price series an absolute value which by itself is not of much importance for the next day direction of the stock.

Before this paper gets deeper into the optimization and mathematical approach of SVM algorithm, we need to understand the theoretical framework associated with it.

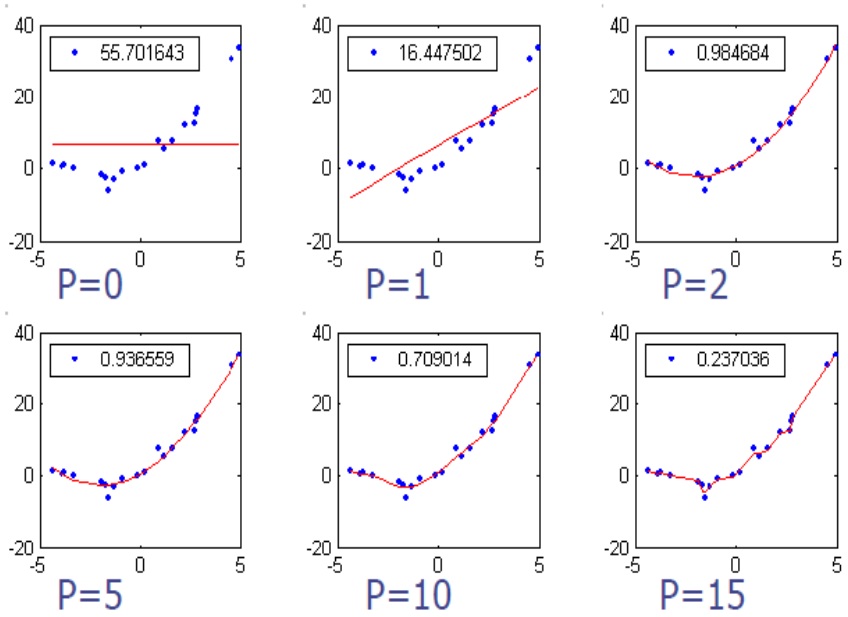
3.4.1 Empirical Risk Minimization:

In all forms of data fitting, the study is trying to find the best fit for our data during the back testing, in order to have better forecast quality for future data points. A particular measure of fitness of our model would be some loss function calculation over all the training data points available for us. We might want to keep aside a portion of data points in order to perform testing on the fitted model which would give us a view over out of sample performance.

A simple approach would be to follow a linear regression model, where we take our loss function as the MSE (mean square error). A closer look would warrant questions regarding the appropriateness of this loss particular function. In fact it could be shown that if the data point distribution is normal Gaussian, then a maximum likelihood fit for IID data points would give us a MSE loss function.

A problem with the above approach is that given a limited number of training data points, we could always achieve a perfect mean square error if we go for a sufficiently higher order polynomial (linear regression models are linear in co-efficient but the predictors can be of higher order) As shown in Figure 3.4 below, we can see that as we increase the order of the fitted polynomial ‘P’, there is a corresponding decrease in the MSE. Total number of blue points in the graphs below is 20 which would mean that a 20^{th} order polynomial would give us a perfect fit according to MSE loss function.

Figure 3.4:



Another problem with the above methodology is that we are not sure about the distribution of data points. It would be a mistake to assume the distribution of our outcome (+/-1 market direction) with the given predictors ROC, stochastic oscillator and close value location as normal density. The fact is that we are not sure about the relationship between the predictors and outcomes in this case. Hence we cannot assume a linear model and do the same analysis as we did in the first part of our trading strategy which was the regression of individual stock series with the components of PCA. It is by the very definition of PCA which guarantees the applicability of regression model there.

Now, the task is to find suitable probability density function (pdf) (p(x,y) and a loss function corresponding to the pdf of data.

$$R(\theta) = E_P\{L(x,y,\theta)\} = \int_{X\times Y} P(x,y)L(x,y,\theta)\,dx\,dy \quad \ldots\ldots\ldots\ldots (3.20)$$

where:

E_p = expected probability.

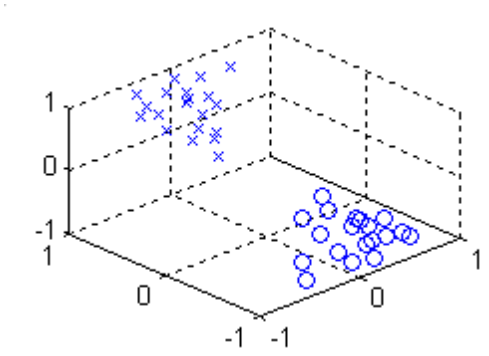
It is by the nature of the our forecasting problem that we cannot observe the p(x,y) function, except for the training data set. This leads to empirical risk minimization approach, where frequently we run into the danger of over or under fitting or over fitting.

3.4.2 Regression for Classification:

Regressions could be used for classification purpose as well, the usual process is to fit the data and use a suitable decision rule to classify the points which are on either side of the regression line/plane.

For example in Figure 3.5 below, we are interested in obtaining the classification of x marked points (+1) and circle marked points (-1). We perform a simple linear regression and fit a plane on these data points.

Figure 3.5:



Projection of the plane thus obtained onto the 2-D surface would be a line as shown in Figure 3.6 below. A classification decision would be based on the location of the data point with respect to this line.

E.g.

Plane: ax+by+cz=0

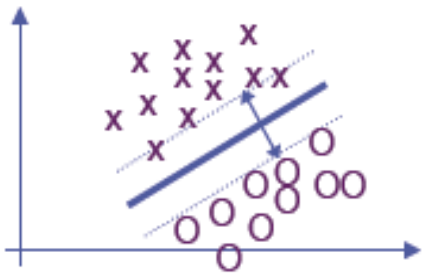
Line: ax+by=0

For a data point x (i), y (i):

ax(i)+by(i)>0 => decision = +1

ax(i)+by(i)<0 => decision = -1

Figure 3.6:



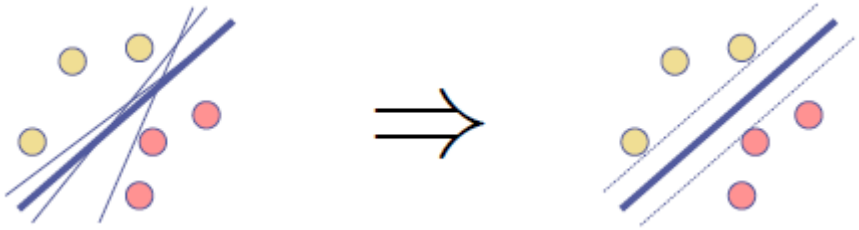
3.4.3 Structural Risk Minimization:

It is easy to see the problem with the classification approach (empirical risk minimization) as described in the previous sections. We won't be getting the ideal line

for our classification purposes as over fitting and under fitting would result in the modeling of noise in data instead of the true structure/probability distribution of the data.

Further, if we have skewed or unsymmetrical observation of the data points corresponding to the +/-1 points or any outlier points lying deep inside of any one of the region, then the resultant plane would be skewed too. The effect of all these shortcomings would be that we would observe a relatively unstable and less efficient classification boundary line as shown in the left side of Figure 3.7 below. While it is easy to see intuitively that a better classification decision would be to have a line which divides the boundary/separation region equally hence maximizing the margin between decision line and the classification regions, as shown in the right hand side of the figure.

Figure 3.7:



This maximization of the margin at the boundary of the two classes of data points would be the approach that would give us a better generalization of our model to the out of sample data points, better than any other form of empirical loss functions.

Apart from this intuitive explanation of the optimization goal of maximizing the margin, there is detailed theoretical base for linking the maximization of margin with the better

generalization i.e. less out of sample error. The concept of VC dimension is introduced below regarding the same.

3.4.3 VC (Vapnik–Chervonenkis) Dimension:

For any classification problem, the VC dimension of a certain classifier is defined as the maximum number of data points which can be shattered (i.e. any possible labeling scheme of the data point could be achieved by the classifier, when the number of points are less than the VC dimension)

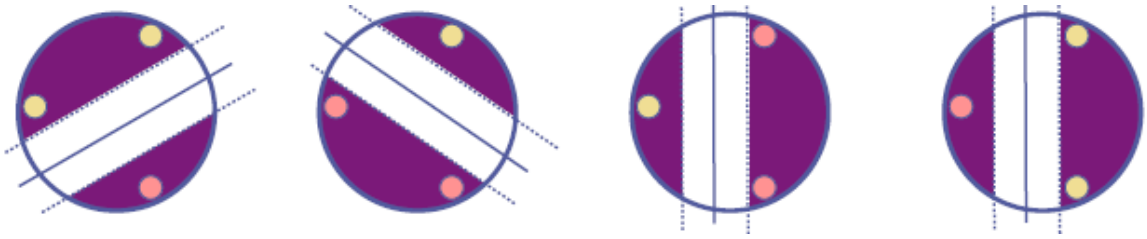
If we have higher VC dimension, it means there are more number of probable models, which can explain the observed data points. The goal here is to reduce VC dimension and in turn get better generalization of the selected model. In general the risk function can be bounded by the out of sample error using VC-dimension h as:

$$R(\theta) \leq J(\theta) = R_{emp}(\theta) + \frac{2h\log\left(\frac{2eN}{h}\right) + 2\log\left(\frac{4}{\eta}\right)}{N}\left[1 + \sqrt{1 + \frac{NR_{emp}(\theta)}{h\log\left(\frac{2eN}{h}\right) + \log\left(\frac{4}{\eta}\right)}}\right]$$

…… (3.21)

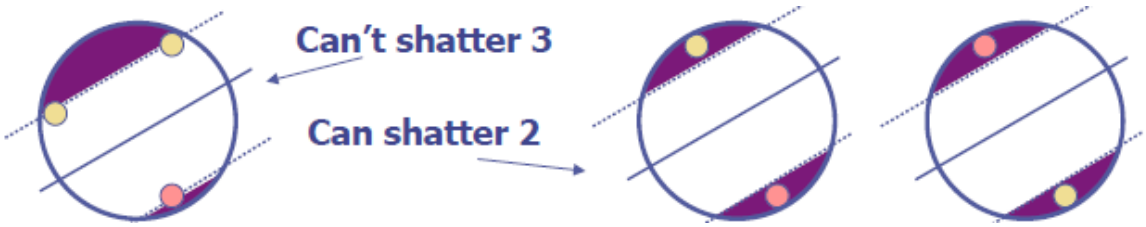
Thus we need to achieve the minimum VC dimension possible. Consider two scenarios where we vary the margin requirement during the classification. We can shatter a large number of points and hence obtain a higher VC dimension if we reduce the margin, as shown in Figure 3.8.

Figure 3.8:



But if for the same data points we increase the margin it's not possible to shatter the three points.

Figure 3.9:



Now we can shatter only 2 points whereas earlier with a lesser margin we were able to shatter 3 points.

This shows that our goal of maximizing the margin while keeping the correct classification for all the data points in the training would indeed correspond to an increase in generalization capability of the model.

3.4.4 Optimization Steps:

Now that we have decided to switch our goal from reducing the mean square risk minimization to maximizing margin given the training data set, we can express the requirement mathematically. Suppose we define the classifier as:

$$w^T x + b = 0 \quad \text{..... (3.22)}$$

where:

w= weight of each classifier.

b= bias.

We have to find the weights w and bias b such that we have the correct classification in the training data set:

$$w^T x_i + b \geq +1$$
$$w^T x_i + b \leq -1$$

This can be expressed as a combined equation while using the labels y:

$$y_i\left(w^T x_i + b\right) - 1 \geq 0 \quad \text{..... (3.23)}$$

In order to maximize margin we will minimize:

$$\min \tfrac{1}{2}\left\|w\right\|^2$$

Since we have a function to minimize while satisfying inequality constraints, we can use the method of Lagrange multipliers to perform optimization:

$$L_P = \tfrac{1}{2}\left\|w\right\|^2 - \sum_i \alpha_i \left(y_i\left(w^T x_i + b\right) - 1\right) \quad \text{..... (3.24)}$$

This can be differentiated w.r.t w and b to obtain the following:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \text{..... (3.25)}$$

Package quadprog() in MATLAB is used to solve this dual optimization problem. The solution obtained through the above optimization would be sparse i.e. that most of the values of alpha would be zero while only significant alpha would correspond to support vectors (data points) at the classification boundary. Hence, these are the points at the boundary that decides the shape of the classification decision function.

3.4.5 Non-Linear classification using SVM:

The method of linear classification by SVM could be upgraded to perform nonlinear classification simply by replacing the higher order function for predictors, this paper uses a radial basis function to capture the non-linearity.

$$k\left(x, \tilde{x}\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|x - \tilde{x}\right\|^2\right) \quad \text{.....(3.26)}$$

Chapter 4: Results

4.1 Results for PCA Strategy:

Stock data of 20 stocks from the XLF financial sector are chosen for our principal components analysis:

ACE, AFL, AIG, AMT, AXP, BAC, BK, BLK, BRK.B, C, COF, GS, JPM, MET, PNC, PRU, SPG, TRV, USB, WFC.

Once these stocks were selected, Statistical arbitrage was performed on the first 10 stocks with systematic returns as the projection of original 20 stocks' returns onto the Eigen vectors and dynamically varying the Principal Components depending on the required amount of variance.

Below are some of the statistics of the trading strategy used for this study:

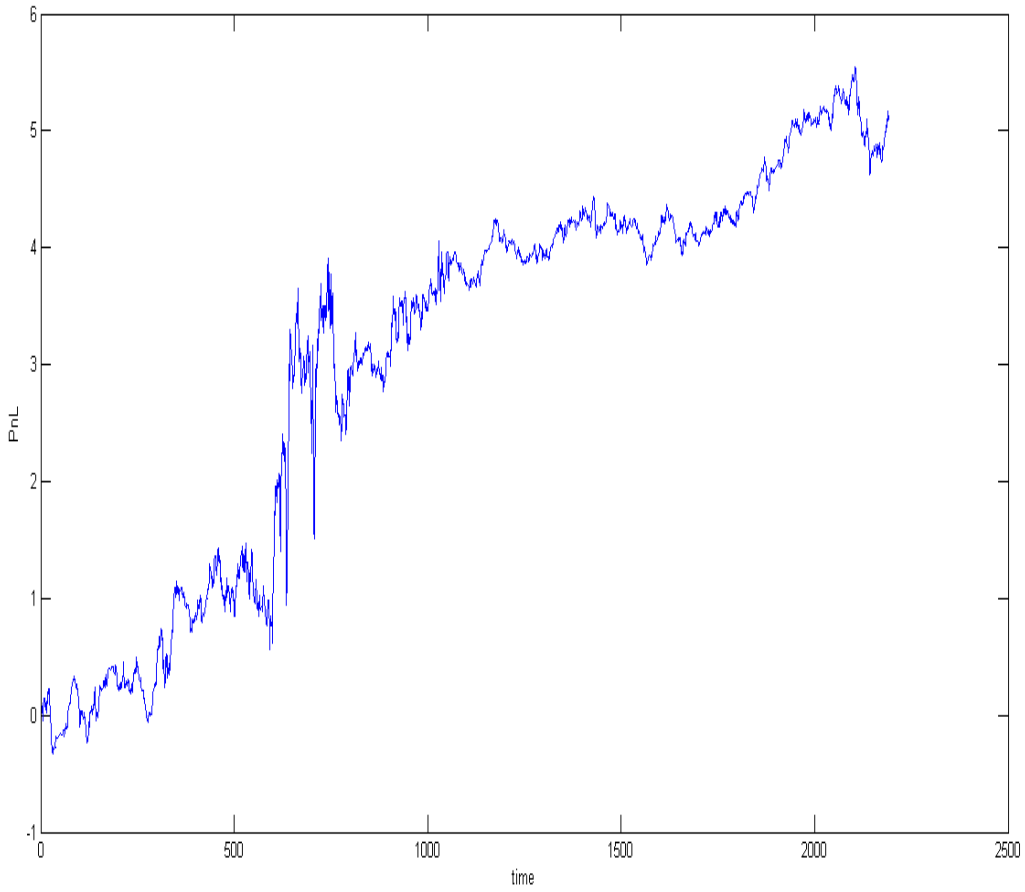
```
total trades =  1596
winners =  896
losers =  691
best winner =  0.5872
worst loser = -0.9831
average holding time =  9.7450
net profit/worst dd =  2.3787
```

The results in below Figure 4.1 show the PnL versus Time graph. PnL refers to the daily change to the value of the trading positions.

PnL = Value today minus value yesterday.

The graph in Figure 4.1 captures the evolution of PnL (profit and loss) of our strategy.

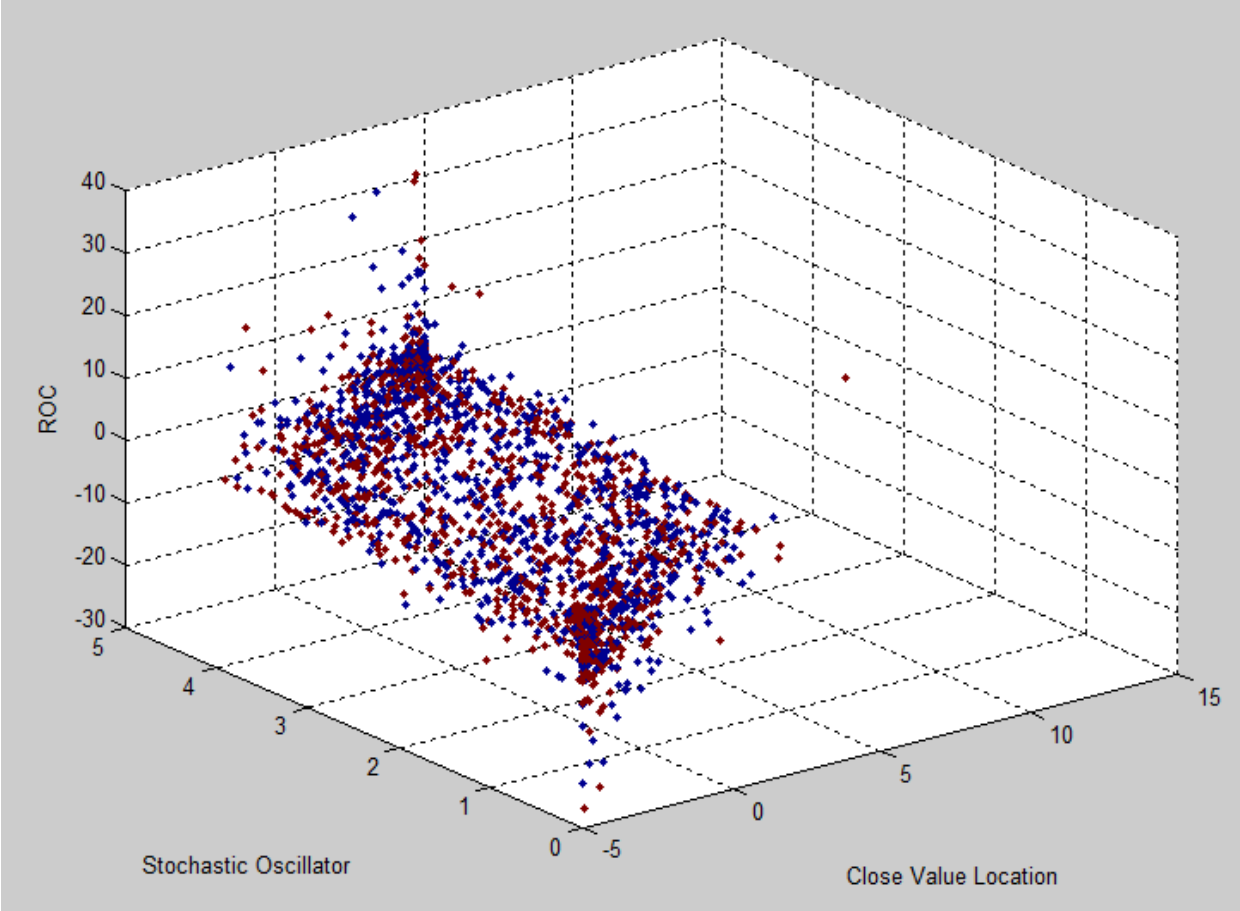
Figure 4.1:



4.2 Performance with SVM Validation:

Classification training data set and out of sample testing is shown below in Figure 4.2 for one of the stock (ACE).

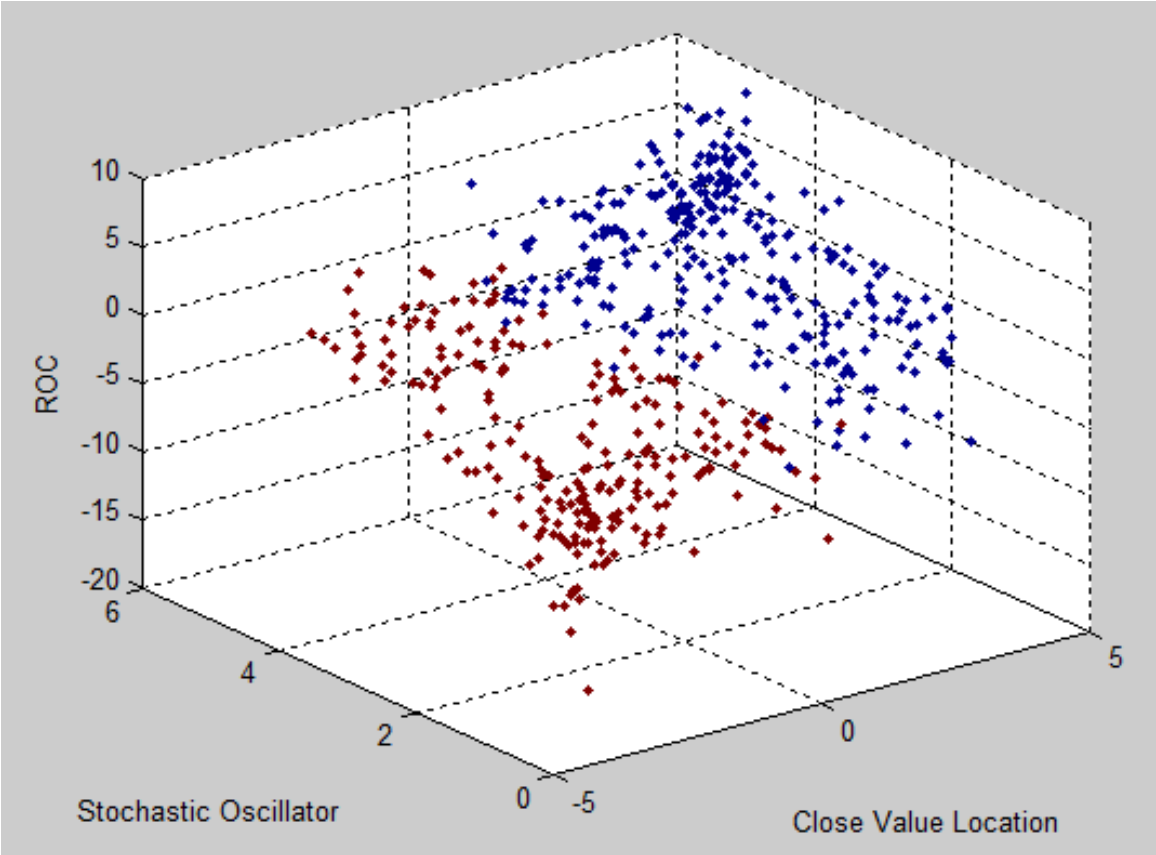
Figure 4.2:



Where blue points are for +1 i.e. upward stock direction and red points correspond to downward market direction. Three technical indicators which are used as the predictors of the stock direction are shown on X, Y and Z labels.

Out of sample testing is shown in below Figure 4.3, we can see that data has been classified nearly linearly; however there is some non-linearity in the classification decision as observed in the intermixing of blue and red points at the boundary region.

Figure 4.3:



In order to fit the parameters for SVM training we are varying the sigma parameter for the radial basis function. The best accuracy is obtained with the sigma and C parameter as shown in the tables below:

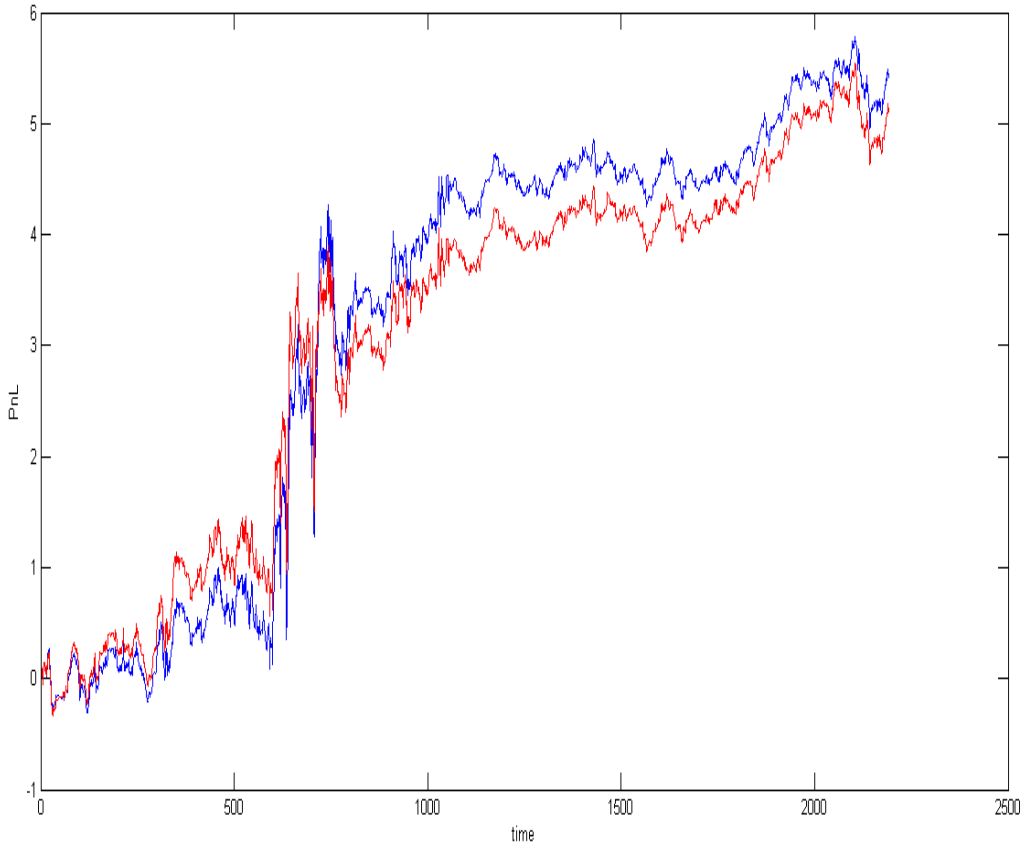
(Sigma)^2=75	
C =	Accuracy
1	56.78
10	55.43
30	57.23
50	58.91
75	56.80
100	57.45

(Sigma)^2=50	
C=	Accuracy
1	54.38
10	55.58
30	56.71
50	56.82
75	55.55
100	57.21

P&L graph of the combined strategy is shown below:

```
total trades =  1519
winners =  856
losers =  654
best inner =  0.4811
worst loser =  -1.0996
average olding tme =  9.8532
net profit/worst dd =   1.8927
```

Figure 4.4:



Chapter 5: Conclusions & Feasible Future Enhancements

It is clear from the PnL graph of the combined SVM+PCA strategy that given a volatile market scenario the combined approach will produce a higher return, whereas under normal conditions, the PCA-only approach would produce a higher return, but with increased volatility. This observation could be attributed to the fact that with SVM validation, we are going to execute fewer trades than we would have otherwise. Hence, we are avoiding additional risk as well as the corresponding additional returns.

A possible list of improvement in the current model and the future work is written below:

1) Stock Selection: An additional constraint on the PCA algorithm to only select the stocks which can be explained by increasingly fewer numbers of common factors/components would lead to better application of statistical arbitrage principles.
2) Adaptive trading rules: A mathematical framework of probability/confidence level indicated by mean reversion signal should be developed. An adaptive strategy would be to compare this probability with the SVM confidence level at the initiation as well as holding of the trade position at the end of every day.
3) Different machine learning algorithms: Other machine learning algorithms which are suited for classification approaches could be applied to check the applicability e.g. neural networks.

4) More Technical Indicators: A large number of the technical indicators had been excluded from the analysis in this paper, inclusion of these indicators are bound to produce higher forecast accuracy.

References

Associated Press, Quant funds endure August turmoil. The Motley Fool, December 6, 2007.

Avellaneda,M., & Lee,J.-H.(2008). Statistical Arbitrage in the U.S. Equities Market. Social Science Research Networks.

Gatev,E., Goetzmann, W.N., & Rouwenhorst,K.G. (2006). Pairs Trading: Performance of a Relative Value Arbitrage Rule. Social Science Research Networks.

Kyoung-jae,K. (2003). Financial time series forecasting using support vector machines. Science Direct.

Barr, A., Quant quake shakes hedge-fund giants Goldman, Renaissance, AQR see losses, but also sense opportunity, Marketwatch, August 13, 2007.

Cont, R., Da Fonseca, J., Dynamics of implied volatility surfaces. Quantitative Finance, 2002, Vol 2, No 1, 45-60.

Bakshi, G. and Z. Chen, 1997, "Stock Valuation in Dynamic Economies," working paper, Ohio State University.

D'Avolio, G., 2002, "The Market for Borrowing Stock," Journal of Financial Economics, 66, 271-306.

Bossaerts, P., 1988, "Common Nonstationary Components of Asset Prices," Journal of

Economic Dynamics and Control, 12, 347-364.

Bossaerts, P. and R. Green, 1989, "A General Equilibrium Model of Changing Risk Premia: Theory and Evidence," Review of Financial Studies, 2, 467-493.

Chen, J., H. Hong and J. Stein, 2002, "Breadth of Ownership and Stock Returns," Journal of Financial Economics, 66, 171-205.

Chen, Z. and P. Knez, 1995, "Measurement of Market Integration and Arbitrage," Review of Financial Studies, 8, 287-325.

Carhart, M., 1997,"On Persistence in Mutual Fund Performance," Journal of Finance, 52, 57- 82.

Connor, G. and R. Korajczyk, 1993, "A Test for the Number of Factors in an Approximate Factor Model," Journal of Finance, 48, 1263-1291.

Conrad, J. and G. Kaul, 1989, "Mean Reversion in Short-horizon Expected Returns," Review of Financial Studies, 2, 225-240.

Engle, R. and C. Granger, 1987, "Co-integration and Error Correction: Representation, Estimation and Testing," Econometrica, 55, 251-276.

Fama, E. and K. French, 1996, "Multifactor Explanations of Asset Pricing Anomalies," Journal of Finance 51, 131-155.

Froot, K. and E. Dabora, 1999, "How are Stock Prices Affected by the Location of Trade?," Journal of Financial Economics, 53, 189-216.

Geczy, C., D. Musto and A. Reed, 2002, "Stocks are Special Too: An Analysis of the Equity Lending Market," Journal of Financial Economics 66, 241-269.

Goetzmann W., et al., "Sharpening Sharpe Ratios", working paper, Yale School of Management.

Hansell, S., 1989, "Inside Morgan Stanley's Black Box," Institutional Investor, May, 204.

Ingersoll, J., Jr., 1987, Theory of Financial Decision-Making, Rowman and Littlefiled, New Jersey.33

Jagannathan, R. and S. Viswanathan, 1988, "Linear Factor Pricing, Term Structure of Interest Rates and the Small Firm Anomaly," Working Paper 57, Northwestern University.

Jegadeesh, N., 1990, "Evidence of Predictable Behavior of Security Returns," Journal of Finance, 45, 881-898.

Appendix A: Code for SVM Learning

```
function statarb()

% READ THE INDUSTRY SECTOR PRICES
slf= xlsread('XLF Prices.csv');

l = 2491;
n = 20;
% READ ALL THE STOCK PRICES REQUIRED FOR PCA
matfiles = dir(fullfile('C:\Users\Gopal\Desktop\PROJECT\data','*.csv'));

data = cell(1,n);

for i=1:n
   str = strcat('data\',matfiles(i).name);
%     str
   data{i} = xlsread(str);
   data{i} = data{i}(2:l+1,:);
end

close_prices = zeros(l,n);
hi_prices = zeros(l,n);
lo_prices = zeros(l,n);

% SORT THE DATA (INITIAL DATA IS IN REVERSE ORDER)
for i=1:n
   close_prices(:,i) = fliplr(data{i}(:,6));
   hi_prices(:,i) = fliplr(data{i}(:,3));
   lo_prices(:,i) = fliplr(data{i}(:,2));
end

% EXTRACT THE CLOSE PRICES FROM THE MATRIX
xlfclose = fliplr(slf(:,6));
xlflo = fliplr(slf(:,3));
xlfhi = fliplr(slf(:,2));

% MATRIX TO STORE SVM OUTPUT FOR DIFFERENT STOCKS AS WELL AS
ETF
svm_output = randi([0,1],l,n+1);

% ASSUMING 58% ACCURACY FOR SVM, RANDOMLY MAKE A CORRECT
DECISION WITH
% PROBABILITY WITH 0.58
for i=2:l
   for j=1:n
```

```
    if(close_prices(i,j) - close_prices(i-1,j) > 0)
        if(rand()>0.42)
            svm_output(i-1,j) = 1;
        else
            svm_output(i-1,j) = -1;
        end
    else
        if(rand()>0.42)
            svm_output(i-1,j) = -1;
        else
            svm_output(i-1,j) = 1;
        end
    end
    svm_output(i,j) = 1;
    end
end

% IF SUM OF DIRECTIONS OF STOCK PRICES IN A SECTOR EXCEEDS 0 WE
ASSUME THAT
% THE SECTOR PRICE GOES UP AND VICE-VERSA
for i=1:l
sum = 0;
for j=1:n
    sum = sum+svm_output(i,j);
end
if(sum >0)
    svm_output(i,n+1) = 1;
else
    svm_output(i,n+1) = -1;
end
end

% CALCULATING STOCK RETURNS
stockreturns = zeros(l-1,n);
for i=1:n
    stockreturns(:,i) = (close_prices(2:l,i) - close_prices(1:l-1,i))./close_prices(1:l-1,i);
end

% CALCULATING XLF RETURNS
xlfreturns = (xlfclose(2:l) - xlfclose(1:l-1))./xlfclose(1:l-1);

window = 60;
th_bo = 1.25;
th_so = 1.25;
th_bc = 0.75;
th_sc = 0.5;
```

```
% PERFORM TRADE BASED ON PCA AND SVM
[total_pnl,pnl,residual_series,total_trades,winners,losers,best_winner,worst_loser,averag
e_holding_time,avg_returns,vol_returns] =
trade(stockreturns,xlfreturns,window,th_bo,th_so,th_bc,th_sc,1,svm_output,1);
plot(total_pnl(300:l-1),'b');
total_trades
winners
losers
best_winner
worst_loser
average_holding_time
np_wd = (total_pnl(l-1) - total_pnl(300))/get_dd(total_pnl);
np_wd

avg_returns
vol_returns
sr = avg_returns/vol_returns;
sr

% RESULTS FOR THE TRADE
% total_trades =  1519
% winners =  856
% losers =  654
% best_winner =  0.4811
% worst_loser =  -1.0996
% average_holding_time =  9.8532
% np_wd =  1.8927
% avg_returns =  0.8490
% vol_returns =  1.7362
% sr =  0.4890

hold on;
% PERFORM TRADE BASED ONLY ON PCA
[total_pnl,pnl,residual_series,total_trades,winners,losers,best_winner,worst_loser,averag
e_holding_time,avg_returns,vol_returns] =
trade(stockreturns,xlfreturns,window,th_bo,th_so,th_bc,th_sc,1,svm_output,0);
plot(total_pnl(300:l-1),'r');
total_trades
winners
losers
best_winner
worst_loser
average_holding_time
np_wd = (total_pnl(l-1) - total_pnl(300))/get_dd(total_pnl);
np_wd
```

```
avg_returns
vol_returns
sr = avg_returns/vol_returns;
sr

% RESULTS FOR THE TRADE
% total_trades =  1596
% winners =  896
% losers =  691
% best_winner =  0.5872
% worst_loser = -0.9831
% average_holding_time =  9.7450
% np_wd =  2.3787
% avg_returns =  0.8191
% vol_returns =  1.7101
% sr = 0.4790

end

% THIS FUNCTION RETURNS THE MAX DRAW DOWN VALUE
function dd = get_dd(pnl)
max = pnl(1);
min_dd = 0;
   for i=2:length(pnl)
      if(pnl(i) < max)
         if(pnl(i) - max < min_dd)
            min_dd = pnl(i) - max;
         end
      else
         max = pnl(i);
      end
   end
   dd = abs(min_dd);
end


% THIS IS THE TRADING FUNCTION THAT TAKES CARE OF PCA|SVM
TRADING
% PARAMS: stockreturns - Different stock returns,syst_returns - the
% systematic returns (with which Principal components are found),window -
% the trailing window size,th_bo - buy to open threshold,th_so - sell to
% open threshold,th_bc - buy to close,th_sc - sell to close,method - static
% pca/dynamic pca,svm_output - the direction of prices given by svm,svm -
% flag whether to do svm or not
```

```
function
[total_pnl,pnl,residual_series,total_trades,winners,losers,best_winner,worst_loser,averag
e_holding_time, avg_returns,vol_returns] =
trade(stockreturns,syst_returns,window,th_bo,th_so,th_bc,th_sc,method,svm_output,sv
m)

num_stocks = length(stockreturns(1,:));
series_length = length(stockreturns(:,1));

% Stats for individual stocks
pnl = zeros(series_length,num_stocks);
residuals = zeros(series_length,num_stocks);
trades = zeros(series_length,num_stocks);

position = cell(num_stocks);
b = zeros(num_stocks,num_stocks);

% Stats for all stocks
total_pnl = zeros(series_length,1);

total_trades = 0;
win_lose = zeros(num_stocks,1);
holding_time = zeros(num_stocks,1);
average_holding_time = 0;
winners = 0;
losers = 0;
best_winner = -1000;
worst_loser = 1000;
entry = 0;
exit = 0;
daily_returns = zeros(series_length,1);

svm_length = length(svm_output(1,:));

   for i=300:series_length
      % THE PCA CASE
      if(method ~= 2)
         syst_returns_prev_yr = get_pca_returns(syst_returns(i-250:i,:),method);
         syst_returns_prev_window = syst_returns_prev_yr(252-window:251,:);
      else
         % OPTIONAL ETF CASE - NOT USED
         syst_returns_prev_window = syst_returns(i-window:i-1,:);
      end
      for j=1:num_stocks
         % MAKE A DECISION FOR A GIVEN STOCK
```

```
        [signal,beta,residual] = generate_signal(stockreturns(i-window:i-
1,j),syst_returns_prev_window,th_bo,th_so,th_bc,th_sc,position{j});
        residuals(i,j) = residual;
        % IF THERE IS NO TRADE SIGNAL
        if(strcmp(signal,'#') || strcmp(signal,'nosignal'))
           if(strcmp(position{j},'so'))
              % IF WE WERE IN SELL TO OPEN TRADE ALREADY
              l1 = length(syst_returns_prev_window(1,:));
              sum1 = 0;
              sum2 = 0;
              for k=1:l1
                 sum1 = sum1 + b(j,k)*syst_returns_prev_window(window,k);
                 sum2 = sum2 + syst_returns_prev_window(window,k);
              end
              % PNL AND RETURNS CALCULATION
              pnl(i,j) = pnl(i-1,j) -stockreturns(i,j) + sum1;
              daily_returns(i) = daily_returns(i) -stockreturns(i,j) +sum2;

           elseif(strcmp(position{j},'bo'))
              % IF WE WERE IN BUY TO OPEN TRADE ALREADY
              l1 = length(syst_returns_prev_window(1,:));
              sum1 = 0;
              sum2 = 0;
              for k=1:l1
                 sum1 = sum1 + b(j,k)*syst_returns_prev_window(window,k);
                 sum2 = sum2 + syst_returns_prev_window(window,k);
              end
              % PNL AND RETURNS CALCULATION
              pnl(i,j) = pnl(i-1,j) +stockreturns(i,j) - sum1;
              daily_returns(i) = daily_returns(i) +stockreturns(i,j) -sum2;

           else
              pnl(i,j) = pnl(i-1,j);
           end

        else
           % THERE WAS A TRADE SIGNAL
           pnl(i,j) = pnl(i-1,j);
           % IF IT WAS A CLOSE TRADE SIGNAL or IF IT WAS AN OPEN
TRADE
           % SIGNAL AND SVM DOESN'T COMPLT WITH THE MEAN
REVERSION,
           % WAIT FURTHER TO ENTER or ELSE ENTER AND TAKE POSITION
           if(~svm || strcmp(signal,'bc') || strcmp(signal,'sc') || (strcmp(signal,'so') &&
~(svm_output(i,j) == 1 && svm_output(i,svm_length) == -1)) || (strcmp(signal,'bo') &&
~(svm_output(i,j) == -1 && svm_output(i,svm_length) == 1)))
```

```
            % TAKING POSITION
            position{j} = signal;

            for k=1:length(beta)
               b(j,k) = beta(k);
            end
            trades(i,j) = residual;

          % CALCULATING DIFFERENT STATISTICS LIKE
WINNERS,LOSERS,TOTAL TRADES etc.
          if(strcmp(position{j},'so') || strcmp(position{j},'bo'))
              win_lose(j) = pnl(i,j);
              total_trades = total_trades+1;
              holding_time(j) = i;
              if(exit==0)
                 entry = i;
              end
          elseif(strcmp(position{j},'sc') || strcmp(position{j},'bc'))
             average_holding_time = average_holding_time + (i-holding_time(j));
              if(pnl(i,j) > win_lose(j))
                 winners = winners+1;
                 if(pnl(i,j) - win_lose(j) > best_winner)
                    best_winner = pnl(i,j) - win_lose(j);
                 end
              else
                 losers = losers+1;
                 if(exit == 0)
                 exit = i;
                 stock = j;
                 po = position{j};
                 end
                 if(pnl(i,j) - win_lose(j) < worst_loser)
                    worst_loser = pnl(i,j) - win_lose(j);
                 end
              end
          end
          end
        end

      end

      % CALCULATING TOTAL PNL
      for j=1:num_stocks
         total_pnl(i) = total_pnl(i) + pnl(i,j);
      end
```

```
    end
    % AVERAGE RETURNS AND VOLATILITY
    avg_returns = mean(daily_returns(300:series_length))*252;
    vol_returns = std(daily_returns(300:series_length))*sqrt(252);

    % AVERAGE HOLDING TIME
    average_holding_time = average_holding_time/total_trades;
    residual_series = residuals;

end
% FUNCTION TO GENERATE TRADE SIGNAL
function [signal,beta,residual] = generate_signal(r_stock, syst_returns,
th_bo,th_so,th_bc,th_sc,position)
    if(length(r_stock)~=length(syst_returns(:,1)))
        signal = '#';
        return;
    end
    ones_array = ones(length(syst_returns(:,1)),1);
    % PERFORM THE REGRESSION TO ESTIMATE BETAS
    [p,bint,r] = mvregress([ones_array,syst_returns],r_stock);
    l = length(r_stock);
    beta = p(1:length(p));

    r_stock_res = r;


    x = zeros(l,1);
    x(1) = r_stock_res(1);
    % CONSTRUCT AUXILIARY SERIES THAT CORRESPONDS TO OU
PROCESS
    for i=2:l
        x(i) = x(i-1)+r_stock_res(i);
    end
    residual = x(l)-x(l-1);

    m = mean(x);
    % AR(1) MODEL ESTIMATES OF RESIDUALS
    [model,e] = arcov(x-m,1);
    % COMPUTING OU PARAMETERS AND THEREBY S-SCORE
    b = -model(2);

    sigma_eq = sqrt(e/(1-b*b));
    s_score = -m/sigma_eq;
    if(s_score < -th_bo && ~strcmp(position,'bo'))
        signal = 'bo';
```

```
    elseif(s_score > -th_sc && strcmp(position,'bo'))
        signal = 'sc';
    elseif(s_score > th_so && ~strcmp(position,'so'))
        signal = 'so';
    elseif(s_score < th_bc && strcmp(position,'so'))
        signal = 'bc';
    else
        signal = 'nosignal';
    end
end
% THIS FUNCTION DOES THE PRINCIPAL COMPONENT ANALYSIS TO
ESTIMATE
% SYSTEMATIC RETURNS
function pca_returns = get_pca_returns(stockreturns,dynamic_pca)
    [coeff, score, latent] = princomp(stockreturns);
    if(~dynamic_pca)
        % STATIC PCA - TAKE JUST 15 FACTORS ALL THE TIME
        pca_returns = score(:,1:15);
    else
        % DYNAMIC - TAKE VARIABLE NUMBER OF FACTORS DEPENDING ON
AMOUNT OF
        % VARIANCE EXPLAINED - 70%
        sig_eigen_vec = cumsum(latent)./sum(latent);
        for i=1:length(sig_eigen_vec)
            if(sig_eigen_vec(i) >= 0.70)
                pca_returns = score(:,1:i);
                return;
            end
        end
    end
end
```