Stock Trend Evolution

9 April 2020

Agarwal, Taylor; Quelle, Henk; Ryan, Cooper; Mentor: Lanius, Melinda.

1 Background/Introduction

For hundreds of years people have been trying to find new ways to analyze the US stock market to allow for easy access to riches. The market is filled with thousands of stocks all being publicly traded. Some of their movements have hidden connections to other stocks that are not easily identified. Some of the stocks have strong effects over others, some dictate markets as a whole, and some are completely unrelated with one another. In this project we aim to test and discover new ways to better understand trends in the US Stock Market using a recent machine learning technique called Principal Component Analysis. During the course of this project we aim to identify groups of stocks with similar variance, by looking at their dependence upon certain "eigenstocks". These eigenstocks are inextricably bound to the stocks themselves, as each eigenstock can be interpreted as a component of the stocks in the data set. Determining how important a particular eigenstock is to a stock will allow us to group the stock with others of similar importance, thereby isolating groups that rise and fall at similar times. Therefore, by looking at a single stock in these groups of stocks, we can determine the motion of the group as a whole, thus giving the us actionable information for maximizing profits on these stocks.

2 Model

Principal Component Analysis, or PCA, is a linear algebra method that reduces the dimensionality of a matrix through statistical and topological reduction methods. PCA analyzes the relationships between each variable of a matrix and evaluates its importance to the matrix itself. In this way, certain dimensions can be disregarded and the important variables come to light. PCA is very valuable to data science, as it allows data scientists to quickly evaluate multidimensional problems with relative ease. It also is useful in machine learning as an unsupervised technique with only one hidden layer.

Imagine a matrix A, a large matrix in $\mathbb{R}^{m\times n}$ made up of values found by looking at m features across n samples. Each column is a sample vector $\vec{x_i}$ for $i \in [1, n]$ of length m, where i is the column index. So

$$A = \begin{pmatrix} \vec{x}_1 & \vec{x}_2 & ... & \vec{x}_n \end{pmatrix} \tag{1}$$

This will be referred to as the *data matrix* for the remainder of this paper.
PCA consists of 5 steps:

1. Shift the data to the origin

2. Compute the Covariance Matrix C
3. Compute the eigenvalues and eigenvectors of C
4. Select a small number of eigenvectors that are representative of the data
5. Compute the weight vectors necessary to approximate the original data using only the subset of eigenvectors

These steps and the information held within them come as a result of PCA methods explained by Hargreaves and Mani 2015 [2], Zhang and Turk 2008 [4], and Strang 2019 [6]. In the following subsections we will explain steps

2.1 Shift the Data to the Origin

The first step in performing PCA on the data matrix is to find the mean vector

$$\vec{\mu} = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i \tag{2}$$

Subtract $\vec{\mu}$ from each $\vec{x}_i$ in A. Then

$$\hat{A} := \begin{pmatrix}\vec{x}_1 - \vec{\mu} & \vec{x}_2 - \vec{\mu} & ... & \vec{x}_n - \vec{\mu}\end{pmatrix} := \begin{pmatrix}\Phi_1 & \Phi_2 & ... & \Phi_n\end{pmatrix} \tag{3}$$

is the new matrix with shifted columns. This shift centers the data around the origin of $\mathbb{R}^m$. This step shows the numerical stability (extremeness of outliers) and resolution of the data, exposing variation in the data and clear differences between values.

2.2 Compute the Covariance Matrix C

Compute $C = \dfrac{1}{m-1}AA^T$

$$C = \frac{1}{m-1}\begin{pmatrix}\Phi_1\Phi_1^T & \Phi_1\Phi_2^T & ... & \Phi_1\Phi_n^T \\ \Phi_2\Phi_1^T & \Phi_2\Phi_2^T & ... & \Phi_2\Phi_n^T \\ ... & ... & ... & ... \\ \Phi_n\Phi_1^T & \Phi_n\Phi_2^T & ... & \Phi_n\Phi_n^T\end{pmatrix} = \begin{pmatrix}c_{11} & c_{12} & ... & c_{1m} \\ c_{21} & c_{22} & ... & c_{2m} \\ ... & ... & ... & ... \\ c_{m1} & c_{m2} & ... & c_{mm}\end{pmatrix} \tag{4}$$

This matrix is called the *covariance matrix* since each entry C_{ab} is the covariance of two samples $\vec{x}_a$ and $\vec{x}_b$,

$$c_{ab} = \frac{1}{m-1}\sum_{i=1}^{m}\Phi_{ai}\Phi_{bi} = \frac{1}{m-1}\sum_{i=1}^{m}(\vec{x}_{ai} - \vec{\mu})(\vec{x}_{bi} - \vec{\mu}) \tag{5}$$

Each diagonal entry c_{aa} is the variance of the sample $\vec{x}_a$, while each off diagonal entry c_{ab} is the covariance of sample $\vec{x}_a$ and sample $\vec{x}_b$. The covariance matrix shows the relationship between any two samples; amongst other things, the covariance can tell you what happens to one variable when you adjust another.

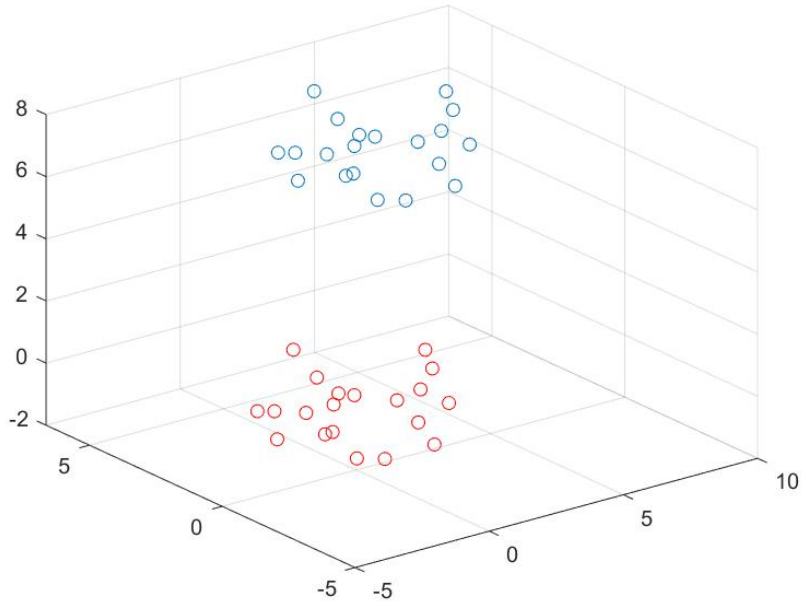

Figure 1: Geometric view of shifting the data by the mean μ. Blue is the original data, and red is the shifted data

A positive covariance between variables tells you that as either sample increases, the other also increases. Oppositely, a negative covariance implies that as one sample increases the other decreases. This is an important step in finding the dimensions that can be reduced, since you can determine how related variables grow and shrink based on a select few variables.

2.3 Compute the Eigenvalues and Eigenvectors of C

To reduce the dimensionality of the C matrix, first find its eigenvalues and normalized eigenvectors. We will call these values λ_j and these column vectors $\vec{v_j}$. Order these eigenvalues such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$ where r is the rank of C. We expect that $r \leq n$ since there will likely be some linearly dependent columns in C for an interacting system of many variables.

The eigenvector associated with λ_1 is called the principal component. Generally, the principal component is the only eigenvector necessary to interpret a large portion of the data. The principal component points in the direction of most variance in the data, so it would hold the most information about the motion of the data than any other eigenvector. More eigenvectors can be held on to as well to interpret more of the data, but the point of PCA is to use as

few eigenvectors as possible to interpret as much data as possible. Once the first k eigenvectors are selected ($k < n$), place them as columns in a matrix V, ordered based on the size of their eigenvalues from greatest to least.

$$V = \begin{pmatrix} \vec{v_1} & \vec{v_2} & ... & \vec{v_k} \end{pmatrix} \tag{6}$$

2.4 Select a Small Number of Eigenvectors that are Representative of the Data

A common method of determining how to reduce the set of eigenvectors is to select the first k eigenvectors that account for a specified percentage of the variance. The percentage of variance accounted for by eigenvector v_i can be determined by λ_i. The percent of variance accounted for by v_i is

$$\text{Percent Explained} = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \times 100\% \tag{7}$$

2.5 Compute the Weight Vectors Necessary to Approximate the Original Data

The selected eigenvectors form the matrix $V \in \mathbb{R}^{m \times k}$. Each vector points in the direction of largest variance that is not already covered by the eigenvectors prior to it, with v_1 pointing in the direction of the absolute largest variance. Each shifted sample vector Φ, as defined in Equation 3, can be projected onto the new space to form a new vector

$$\widehat{\Phi} = \vec{\mu} + \sum_{i=1}^{k} w_i \vec{v_i} \tag{8}$$

where $\widehat{\Phi}$ is the projection of Φ and $\vec{\mu}$ is as in Equation 2. This new vector $\widehat{\Phi}$ is

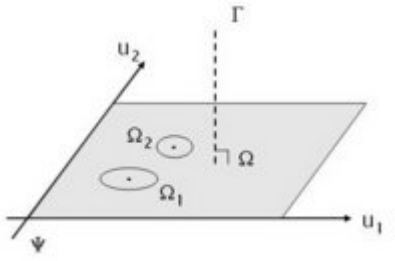

Figure 2: A Visualization of the projection described in Equation 8. Sheng Zhang and Matthew Turk (2008) [5]

formed by a linear combination of only the selected eigenvectors and the average

sample. The weights for a particular combination can be found by

$$w_i = \vec{v_i}^T \Phi \tag{9}$$

Placing these weights into a weight vector Ω yields

$$\Omega = \begin{pmatrix} w_1 \\ w_2 \\ ... \\ w_k \end{pmatrix} \tag{10}$$

and therefore

$$\widehat{\Phi} = \vec{\mu} + V\Omega \tag{11}$$

This weight vector can tell us a great deal about the sample vector. If a particular magnitude of w_i is very large, then you can conclude that the variance in Φ is strongly accounted for by v_i. Each Ω can be placed in the columns of a matrix $W \in \mathbb{R}^{k \times n}$. Then, an approximate recreation of the original samples can be accomplished by calculating

$$A \approx VW \oplus \vec{\mu} \tag{12}$$

where $\oplus$ indicates addition across the columns of VW. If all r eigenvectors are retained in V, then

$$A = VW \oplus \vec{\mu} \tag{13}$$

2.6 Connection Between PCA and Stocks

This project will explore the applications of PCA to the United States Stock Market. In particular this project will identify so-called "eigenstocks" from a sample set of stocks which describe the variance of the stock market as a whole, and attempt to form subsets of the sample set of stocks that have similar variance.

The percentage of total weight devoted to each eigenvector determines how heavily each original sample depends on the selected eigenvectors. The percentage of weight can be computed by

$$\text{Percent of Weight} = \frac{w_i}{\sum_{j=1}^{n} w_j} \times 100\% \tag{14}$$

For example, if the first element of the weight vector Ω for a sample Φ has 20% of its total weight, then 20% of Φ is accounted for by the first eigenvector. Samples with very high weight percentages for a given eigenvector will have similar variance to other such stocks, and therefore groupings of samples with similar variance can be made based on high weight percentages.

The data used for this project will be pulled from a large data set of 7195 stocks dating back to 1960 publicly available on Kaggle (Marjanovic 2017) [3]. This data includes opening and closing values for each day the stocks were

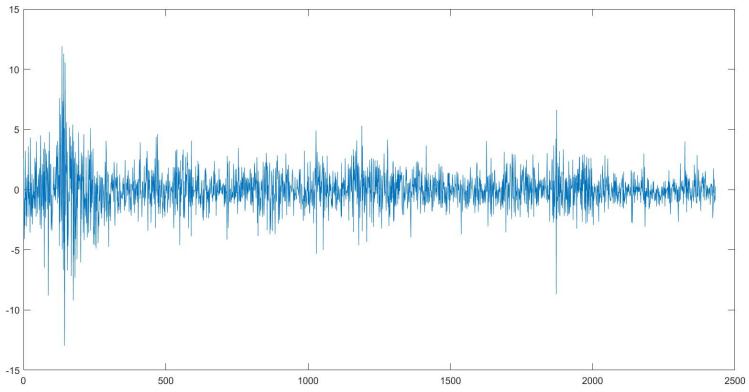

Figure 3: Percent Change of APPL Stock over Time

available for trade. We will look at the percentage change over each day for each stock in its sample set. This can be computed by

$$\text{Percent Change} = \frac{\text{Close} - \text{Open}}{\text{Open}} \tag{15}$$

The choice to use percent change instead of any other metric comes as a result of trying to normalize the importance of all stocks, so that high valued stocks do not hold more weight than lower valued ones. Ultimately, we will perform PCA on a data matrix consisting of stocks as columns, dates as rows, and the percent change for the stock on that day as entries. A sample plot of this data for the APPL stock is shown in Figure 3. We will identify stocks that are similar in variance by comparing their dependence on certain eigenstocks.

3 Future Plans

Our original plan was to find similar trending stocks that can be grouped together using PCA analysis. Moving forward, we will pursue two distinct paths of extension and exploration. Through PCA we can be extend to larger groupings of stocks with more specific related market groups. Exploration entails utilizing other techniques such as CX decomposition that can be implemented to see how accurately specific groups of stocks can predict behaviors of subsections of the market or even behavior of the market as a whole.

3.1 Extension

Through PCA there are many different approaches to this project. So far, the one that we have focused on is gathering 100 of the top Fortune 500 company

stocks (Fortune, 2020) and using PCA analysis to group them into trend related subsections. This allows us to see which of the largest companies trend together. An extension of this approach would be to do similar analysis on different subsections of the market. Technology related stocks could be grouped together and the same analysis could be run to see which stocks in that field trend in a similar pattern and therefore rely on each other. The same thing could be done with the food industry, the retail market, the field of finance, and many others to see which companies in those sectors can be expected to move together. The ideal situation, with unlimited computing power, would be to analyze all stocks in the entire market. This would allow for unobstructed groupings of all companies without bias. However, with the large number of publicly traded companies and the desire to use multiple years of collected data to find accurate trends, the matrix needed for that would be far too large for most computers and software to handle.

3.2 Exploration

While trying to better understand PCA analysis we encountered another type of analysis was discovered called CX decomposition. This form of decomposition entails creating a matrix C from columns in a matrix A then finding a matrix X that when multiplied by matrix C results in a desired error between matrix A and the new combined matrix, CX. With CX decompositions we believe that there are even more routes to be taken to try to better understand the stock market and how the different stocks are related. Time permitting, we will use CX decomposition in two separate ways. The first way is to find the company in a specific industry, such as food, finance, or technology, that has the strongest pull over that section of the market. Using CX decomposition it could be determined at what percent error each stock can predict the rest of the market. The stock with the lowest error in predicting the movements of all other stocks in its respected grouping would have the highest effect on that section of the market. The second way in which CX decomposition can be used is to find small groups of companies in a large section of the market that control the rest of that section. This would be done by lowering the percent error in predicting the section of the market by using additional stocks in our CX decomposition. The first additional stocks used in the decomposition would result in the largest change in error with each new stock that is added contributing less and less to the change in error. A cutoff threshold would be set for the size of the change in error and every stock that resulted in a change in error above that threshold would be considered a more influential stock in that area of the market.

4 Applications of the Model

The application behind Principal Component Analysis (PCA) and the linear algebraic modeling of the stock market is in short, a different perspective of how to view overall trends and variance in the market. The idea of an "eigenstock,"

post PCA, is not that it is necessarily representative of a real stock that someone would go to the market to acquire. Rather, an "eigenstock" can be observed as a close approximation that should reveal similar behavior(Shahnawaz and Ghazanfar 2017). The groupings of these stocks based off of their highest weights of the "eigenstock" in linear combination with other "eigenstocks," will in theory give us stocks that vary in nearly the exact same manner. So, this concept of grouping "eigenstocks" based on their highest "eigenstock" weight, is far more useful than say, using this to identify one or a few stocks that dominate the market. Taking specific groups in association with one-another will allow for an individual to look at the stocks that are related based on its grouping in a more magnified way. Looking at stocks in groups is useful if for example, there is a

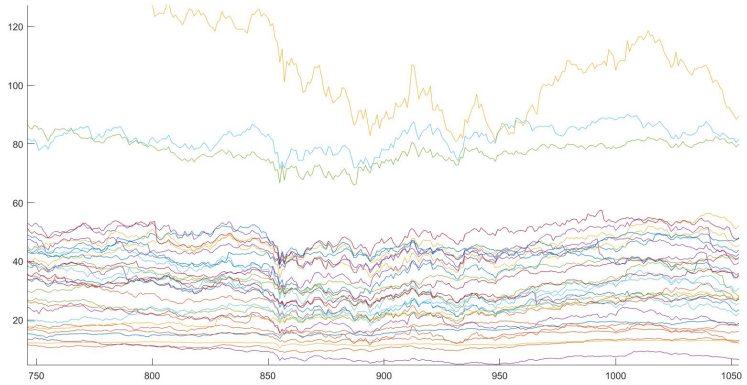

Figure 4: An example of stocks grouped by PCA with similar variance

barrier to enter the grouping based on the highest weight of its "eigenstocks" in linear combination. This should help in some cases, to determine a dominant stock in a grouping, meaning that the common "eigenstock" weight far exceeds the barrier of entry. This will allow us to key in on this stock to determine the overall direction of the portfolio. This is applicable because usually the market investor would like to play it safe and diversify one's portfolio of stock investments, and an individual having the knowledge of which stocks are in association with each-other is a good method to observe when selecting a series of stocks that won't necessarily be heavily dependent on one's entire investment sum.

To summarize the explanation based off of the method selected in the research, the idea is to safely invest a consumer's income into a diverse portfolio of stock investments so that the outcome is profit. This method allows us to group stocks based on their average trends, so in some cases the observer would expect to see stocks of similar industry grouped together but it should be noted that industries in seemingly unrelated fields of business can and will be grouped together. It is important that the user understands that this is a mathematical

model that is heavily dependent on the transformation of raw data. The numbers are what the entirety of this research is based on and all bias to specified stocks is disregarded. There are far too many factors in business that can affect the valuation of a companies share price, so grouping them by how they change is far more efficient to get a relative sense of how certain stocks may trend in the future as well as in the present, so that the investor may make a far more educated decision.

5 MATLAB Code

Code for PCA

```
function [V,E,mu,W] = PCA(data, p)
    % This function performs PCA on the data matrix, where the columns
    % of data are samples and the rows are features
    % Input Arguments:
    % data - The data to be analyzed, m >> n
    % p - the minimum percent of variance to be accounted for
    % Output Arguments:
    % V - The principal components of data
    % E - The eigenvalues associated with each principal component
    % mu - The mean sample
    % W - The weight matrix,
    nsamples = size(data,2);
    nfeatures = size(data,1);
    %% Perform PCA to find eigenvectors
    mu = mean(data,2);
    T = data - mu;
    C = (1/(size(T,1)-1)).*(T*T');
    [V,D] = eigs(C,nsamples);
    E = diag(D);
    explained = E./sum(E);
    tot_explained = 0;
    j = 1;
    V_red = [];
    while tot_explained < p
        v_temp = V(:,j);
        v = v_temp./norm(v_temp);
        V_red = [V_red, v];
        tot_explained = tot_explained + explained(j);
        j = j + 1;
    end
    V = V_red;
    %% Find weight vectors
    W = [];
    for i = 1:size(data,2)
        samp = data(:,i);
        phi = samp - mu;
        omega = V'*phi;
        W = [W, omega];
    end
```

References

[1] Fortune Media. "FORTUNE." Fortune, Fortune, 3 Apr. 2020 `fortune.com`.

[2] Carol Anne Hargreaves, and Chandrika Kadirvel Mani, *The Selection of Winning Stocks Using Principal Component Analysis*, American Journal of Marketing Research, Vol. 1, No. 3, pp. 183-188, (2015)

[3] Marjanovic, Boris. *Huge Stock Market Dataset*. Kaggle, 2017. `https://www.kaggle.com/borismarjanovic/`

[4] Muhammad Waqar, Hassan Dawood,Muhammad Bilal Shahnawaz, Mustansar Ali Ghazanfar, and Ping Guo, *Prediction of Stock Market by Principal Component Analysis*, Proceedings of the 2017 13th International Conference on Computational Intelligence and Security

[5] Sheng Zhang and Matthew Turk. *Eigenfaces*. Scholarpedia, 3(9):4244 (2008).

[6] Strang, Gilbert. *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.