Group 62 - Progress Report

Neill Killgore, Rajiv Kamal, Eissa Qader, Kha Tran, Bryce (Bo) Meyering

4/1/2022

Introduction

Under the STOCK (Stop Trading on Congressional Knowledge) act of 2012, senior US government officials have to report any stock, bond, or commodities transaction within 45 days of the transaction, provided that the total transaction amount is greater than $1000. Recently, news of insider trading and violations of the STOCK act by congressional members has made many headlines in US national news [1]. This type of behavior is troubling due to the timely and, many times, highly sensitive information they are exposed to on a regular basis. Though the purpose of the STOCK act is to prevent our representatives from acting on this information and gaining a distinct advantage in the market, the penalties for not complying are minimal and ineffective (a fee of $200 USD for each transaction reported late). While proving the intent behind a congressional member's trades is a difficult task since we do not know what they know, we thought it would be worthwhile to analyze representatives' trading behavior and transaction timing before major moves in either the market or individual securities they hold.

While there are some methods for detecting insider trading that have been proposed in recent articles[2][3][4], most rely on released litigation records for transactions that could form the basis of a training data for the models they employed. While many congressional members have been implicated in recent allegations of insider trading, there are many legal obstacles to proving the 'materiality' of any information they receive.[5]. Thus no members have yet been charged with any infractions of the law regarding trading securities based on non-public information. To make matters worse, there are several other obstacles to identifying suspicious trading activity in Congress. First, congressional members only have to report their data in discrete bins of the dollar amount of the purchase or sale, not in the number of shares they traded. So the starting data is inherently imprecise. In addition, though persons engaged in insider trades are more likely to trade in options rather than stocks, due to the potential payouts involved[6], congressmen are not required to report whether the transactions were options or stocks.

Main Objectives

For this project then, we have identified three main objectives that will form the basis of our analyses.

- Thus, our first and main research goal is to leverage publicly available daily stock price data and Congressional stock transaction disclosures to identify 'suspicious' trades within Congress. We plan to investigate any correlations between congressional members' trades and price movement in the market following the trades that congressional members make.
- Our Second research goal for the project is to identify the best and worst traders within each respective house of congress or political party by computing their individual returns and portfolio *alphas* compared to broader market indices.
- Our third and final goal is to identify any distinct clusters of congressional members based on similar trading patterns, positions held, cumulative returns, trading dates etc.. We will try to find the variables that are most associated with each cluster see if any congressional members that have suspicious trades cluster together or are distinct from those members with 'typical' portfolios. In general, we will try to identify the data characteristics of outliers in the clustering analysis.

Problem Statement

To state our problem more succinctly:

[1] see (https://www.theatlantic.com/politics/archive/2022/01/congress-stock-trading-ban/621402/) for more reading

[2] Identification of Insider Trading Using Extreme Gradient Boosting and Multi-Objective Optimization. Shangkun Deng et al. Information 2019, 10, 367; doi:10.3390/info10120367

[3] A Deep Learning Based Illegal Insider-Trading Detection and Prediction Technique in Stock Market. Sheikh Rabiul Islam. ArXiv, 2018, abs/1807.00939

[4] How to detect illegal corporate insider trading? A data mining approach for detecting suspicious insider transactions. Intelligent Systems in Accounting, Finance, and Management. Vol 26, Issue 2, 2019

[5] How Senators May Have Avoided Insider Trading Charges. Robert Anello. Forbes https://www.forbes.com/sites/insider/2020/05/26/how-senators-may-have-avoided-insider-trading-charges/?sh=504a1bfb27ba

[6] Early Detection of Insider Trading in Option Markets. Steve Donoho. KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data miningAugust 2004 Pages 420–429https://doi.org/10.1145/1014052.1014100

Recent allegations of congressional insider trading have revealed the inherent difficulties of proving the materiality of any non-public information. Current methods for insider trading detection rely on labeled data and/or litigation records, records that do not exist for members of congress who trade securities. We propose to use public data to calculate portfolio metrics for each congressional member and identify outlier trading activity by clustering in order to flag trades for either suspicious timing or amounts.

Dataset Description

We have identified three separate datasets that we plan to leverage in addressing our research questions.

- The first dataset is a live database containing all of the disclosed stock purchases/sales of all U.S. Congress members. We have accessed this database through a Python API at Quiver Quant for a small fee. Pursuant to the Code of Federal Regulations, government officials are only required to report the value of a sale or purchase within 11 distinct bins and what investment they sold/purchased, but not the quantity or exact price. This leaves some ambiguity in the numerical side of the data that we will have to address by making some broad, less-than-ideal assumptions. Our dataset currently has coverage from 2016 until present for all congressional members who disclosed any transactions.
- In order to supplement the above dataset, we have also pulled daily stock data for all of the unique ticker symbols found in our congressional trading data. This data was pulled from Yahoo Finance using the R package `tidyquant` starting from January 1st, 2016 until present date. We plan to focus most on the `adjusted_close` variable in order to calculate returns.
- Finally, we used curl script GET-requests to obtain metadata for each congressional member using the ProPublica API. This included a variety of information such as Party, gender, congressional committes, etc. We mainly used this data to quickly obtain all of the Party data for each of the 167 unique members in our trading dataset.

The first two datasets are in tidy formats with self explanatory variable names. However, I have listed them below

Congressional Trading Data

- **Report Date:** Date that the transaction was disclosed under the STOCK act.
- **Transaction Date:** Date of the transaction
- **Ticker:** Ticker symbol of the transaction
- **Representative:** Name of the Congress member
- **Transaction:** Type of transaction, 'Sale' or 'Purchase'
- **Amount:** Lower bound of the transaction range
- **House:** Congressional body that the representative is a member of, 'Representatives' or 'Senate'
- **Range:** The range of the transaction value

Here is a glimpse at a few random rows of the congressional trading dataset (only select columns for brevity's sake):

TransactionDate	Ticker	Representative	Transaction	Range
2020-06-09 00:00:00	HXL	Lois Frankel	Purchase	$1,001-$15,000
2019-08-28 00:00:00	LKQ	Susie Lee	Purchase	$1,001-$15,000
2020-08-28 00:00:00	MSFT	David B. Mckinley	Sale	$1,001-$15,000
2021-09-30 00:00:00	MA	Zoe Lofgren	Purchase	$1,001-$15,000

Historical Market Data

- **Symbol:** Stock ticker symbol
- **Date:** Date of the observation
- **Open:** Opening price
- **High:** High price
- **Low:** Low price
- **Close:** Closing price
- **Volume:** The number of shares traded
- **Adjusted:** Closing price after accounting for major corporate actions

symbol	date	open	high	low	close	volume	adjusted
AAPL	2016-05-04	23.8000	23.975	23.4550	23.5475	164102000	21.77234
AAPL	2016-12-06	27.3750	27.590	27.2975	27.4875	104782000	25.84056
AAPL	2016-08-18	27.3075	27.400	27.2550	27.2700	87938800	25.50515

symbol	date	open	high	low	close	volume	adjusted
AAPL	2019-10-10	56.9825	57.610	56.8250	57.5225	113013600	56.43900

The data from ProPublica was a nested JSON file. We used Python to grab the data from the curl output and export it to .csv format to upload in R.

Congressional Member Data

- **Party** (This is the main variable we wanted, though many others were present in the database)

Approach and Methodology

Our approach to analyzing and modeling the data has several parts which are discussed in the sections below. In general, we followed the following schema: Data Cleaning -> EDA -> Clustering -> Calculating Member Returns -> Identifying Suspicious Trades

Data Cleaning and Merging

There were significant nomenclatures inconsistencies that needed to be amended when joining the trading data to the Congressional member data. Around a third of the members either had some misspelling or several versions of their names listed in the datasets. These were corrected through regular expression extraction and or exact name matching. There were also a few issues such as a reporting date and transaction date that got switched, so those were fixed accordingly. We also created a few more variables in the dataset such as `report_lag` (`report_date-transaction_date`) and `overdue` which is the number of days past the 45 day limit that the transaction was reported.

Exploratory Data Analysis

Once we cleaned the data, we first performed some exploratory data analysis to get a general idea of the structure of the data. We created interactive data visualizations using the `plotly` library to look at summaries for the most commonly traded stocks, the stocks with the highest transaction amounts, the congressional members with the highest and lowest reporting lags, and the relationships between all of these variables. We also looked at many of these key variables on a Party basis to see if there were any large differences between Democrats and Republicans. The results from these will be discussed in the preliminary results section below.

Clustering

Using the congressional trades data we then created several sets of sparse matrices to use in cluster analysis. We decided to create matrices in two basic categories: tickers and transaction dates. Focusing in on the first category we created three sparse matrices - a basic matrix with transactional history for each stock, the proportional amount of transactions for each stock, and the proportional transaction amount for each stock. Each of these matrices were 167x818 (Members X Unique Tickers). For the second category of sparse matrices, we pulled the transaction date history for each member and calculated 3 matrices for the total transaction amount per day, transaction count proportion per day, and transaction amount proportion per day for each member. The matrices were filtered to remove all zero columns, i.e. any dates that had no transactions. Both classes of matrices were scaled by selecting indices of the non-zero elements in each column, mean centering and scaling the indexed elements, and overwriting the original matrix values, thus ensuring the data was scaled while all of the zero elements remained 0.

We then worked on reducing data dimensionality by several different PCA algorithms for high dimensional sparse data. Specifically we used a few of the methods in the `sparsepca` package but found them to be computationally slow when solving for more than 20 components. In comparison, we were able to use the `sparsesvd` algorithm from the like-named package to quickly compute the truncated sparse SVD components. We used this to compute the first 20 principal components for each of the sparse matrices to use for clustering. We implemented a spherical k-means algorithm from the `skmeans` package and also the DBSCAN clustering algorithm from the `dbscan` package.

Finally, we computed the cosine similarity between members of each cluster and with every member of each cluster against members of other cluster as a quality check to ensure that the cluster membership did indeed have more similarity to each other than out-of-cluster members.

Calculating Member Returns

Assumptions

Due to the incomplete nature of the data, we had to make some assumptions in order to calculate returns for each member.

- **No member ever has a negative balance of either stocks or cash in their accounts.** In order to accomplish this, we used trades and price data to calculate the ending balance of each member as if they had started with $0 and zero shares. For negative values (e.g. -500 shares AAPL), we made the assumption that the congress member started the trading period with enough of the asset to be able to sell it. In order to adjust the data, the inverse of minimum balance of each asset, if negative, was added to the each member's starting balance.
- **All executed trades occur at the lower bound of the specified range.** The data did not specify the exact number of shares or value of the trade. For consistency between all traded and members, the `Amount` column of the congressional trading data was used to calculate the number of shares purchased or sold.

Calculation Method

To optimize the matrix operations required, we transformed the prices table to a data structure with each column representing a stock ticker and each row represented a date. The cells contained the value from the `adjusted_close` column of the original prices table. We added a `signed_amount` column to the congressional trades dataframe to represent sell orders with a negative number and buy orders with a positive number.

For each member in the dataset, the following operations were performed:

1. Get a list of all stocks traded by the member and first/last trades dates from the congressional trades table.
2. Create new dataframes `member_prices`, `member_orders` containing only data for the relevant member, dates, and stock tickers. Add `CASH` column to `member_prices` to track cash balance.
3. Copy `member_prices` dataframe to a new `member_trades` dataframe. replace all values with 0.
4. For each trade in the `member_orders` dataframe:
    - Calculate number of shares using `signed_amount` column and insert into appropriate row (date) and column (ticker) of `member_trades`.
    - Insert the inverse of the cash value from 4a into the `CASH` column.
5. Create `holdings` dataframe by calculating the cumulative sum of each column of `member_trades` dataframe. The `holdings` dataframe tracks the daily amount of each asset that each member owns. (e.g. 5 shares NFLX, 2 shares AAPL, $1000 cash).
6. Create `holdings_dollars` dataframe by multiplying the `member_prices` dataframe by the `holdings` dataframe to calculate the daily cash value of each asset in the portfolio.
7. Create `portfolio_values` dataframe by summing across each row to calculate the daily total cash value of the portfolio.

Once the daily portfolio cash value was calculated for each member, the `tidyquant` and `PerformanceAnalytics` libraries were used to calculate daily returns and alphas.

Identifying Suspicious Trades

The article by M.Fevsi Esen *et al* [7] suggests capturing returns after a set window of time for each trade. We calculated the return of each trade and the S&P500 index after 30 days. From this data, we can calculate how each trade performed in relation to the overall market.

We can then use a clustering algorithm as suggested by M.Fevsi Esen *et al* to identify suspicious trades. DBSCAN is a useful clustering technique for this, since it labels outliers as "noise".

Preliminary Results

EDA Results

Based on our EDA, we have seen that the vast majority of transactions occur in just a handful of securities such as Apple, Microsoft, and Amazon, with the data roughly following a pareto distribution. This is also true for the securities that are associated with the highest reporting lag, only a few stocks account for the largest average reporting lags. In addition there seems to be no linear relation between the reporting lag for each stock and the transaction volume, however a negative non-linear trend does exist. This also holds when we look at the data on a member basis. Members who trade more frequently are less likely to report trades delinquently, though this is a non-linear relationship. On average, the vast majority of members report trades in a timely manner, but there are many notable exceptions to this. We looked at the transaction count (purchase vs sale) on a daily basis, as well as the total transaction amounts for the same time periods. From this we identified a series of dates where Suzan DelBene made purchases and sales in Microsoft within a few days of each other all for amounts greater than $5m USD. These periods are interesting, especially since her husband was the CDO at Microsoft during that time. Keying in on these and other anomalous periods in the data will help us narrow down the search for suspicious trading.

[7] How to detect illegal corporate insider trading? A data mining approach for detecting suspicious insider transactions. Intelligent Systems in Accounting, Finance, and Management. Vol 26, Issue 2, 2019

Clustering

We used both spherical k-means clustering and DBSCAN on the sparse matrices that we constructed. We believe that DBSCAN yielded the most informative results for us since it creates clusters on a density basis instead of identifying an arbitrary set number of clusters. From the data we know that most members trade very little and trade in a small number of common stocks, thus we have an underlying assumption that the vast majority of members have similar uninteresting trading portfolios and thus should cluster together. Whereas k-means generates labels based on the centroids of the clusters, DBSCAN throws all of the non-clustered points into a 'Noise' category, which is perfect for identifying outlier traders. PCA with the matrices constructed from member portfolio data yielded broad distributions with more noise points when clustered with DBSCAN, However, when we used the transactional time series data from each member to construct sparse matrices, DBSCAN was able to identify a single dense cluster of traders with just a handful of wild outliers. These included members 'David Perdue', 'Josh Gottheimer', and 'Greg Gianforte' in addition to others. While the members of the dense cluster have a high degree of cosine similarity with other members in the dense cluster, outlier traders have low cosine similarity between all other traders including the other outlier traders in the 'noise' group. We plan to improve the clustering as discussed in a a later section.

Member and Trade Returns

Portfolio returns and portfolio alphas for all members in the dataset seem to indicate that congresspeople, do not substantially outperform the market. The largest alphas show the member outperforming the market by single digits over the course of the entire analysis period, and many members under-performed the market. Further investigation is needed to determine whether portfolio returns are a valid predictor of insider trading; however, initial results seem to indicate that they are not.

Thirty day returns from individual trades are far more diverse, with some members, in a single trade, beating the market by more than 100%. The mean and median of the thirty day returns for all purchases are both near 3%, which is double the thirty day market return and may indicate that a significant number trades made by congress are far more profitable than would otherwise be expected. DBSCAN identified a single dense cluster that contained most trades with a few outliers.

The most profitable trade in initial results was made by Brian Mast. He purchased IPWR at the end of December 2020. In January 2021, less than 30 days later, IPWR published a whitepaper describing a new battery charging technique that could be used to improve charging efficiency in electric vehicles. Mast is a member of the transportation committee.

Future Work to be Completed

Optimizing Clustering Algorithms

We plan to continue trying out different methods to cluster the data. Specifically, we need to identify a suitable ϵ value to use for the DBSCAN algorithm. We plan to look at an improvement of DBSCAN called OPTICS which orders the points to identify closest neighbors. There are other variations on this algorithm that are similar, but this should improve our clustering since OPTICS helps identify clusters that have varying density. Depending on our results from the OPTICS algorithm, we may also try a different dimensionality reduction technique, ISOMAP, to create a low dimensional embedding of our data before clustering.

We will also experiment with hierarchical clustering as described in the article by M.Fevsi Esen *et al.*

Identifying the Best Traders

We plan to continue investigating both portfolio statistics and returns from individual trades. We still need to evaluate how timely sales of securities may have protected a member from a market retraction. We will look at the members who beat the market with the highest frequency and the largest returns.

Suspicious Trading

We plan to use density-based and hierarchical clustering with the individual trades data to identify outliers. More work is needed to incorporate trading volume and trade amounts into the clustering algorithm. Both the `Amounts` and `volume` fields in the data have extreme outliers that skew the results and make them unusable. Dimensionality reduction will be necessary for visualization of clusters.

Any outliers identified by the clustering analysis should be further investigated to determine if any news or other events may have impacted the price following the trade. Additionally, committee membership data will be incorporated to determine if the congressperson may have had privileged access to information due to their assigned duties.