Network-based Anomaly Detection for Insider Trading

Sameera Bammidi
George Mason University
sbammidi@masonlive.gmu.edu

Divya Vajja
George Mason University
dvajja@masonlive.gmu.edu

ABSTRACT

Insider trading is illegal when a person buys or sells a company's securities for a profit using crucial non-public information. Tracking such trading activities is hard, and recently data mining techniques are gaining popularity for analyzing different trade patterns and relationships among a range of insiders. Data from U.S. Securities and Exchange Commission (SEC) can be used to gather information about the trading behaviors of different insiders from different companies.

In this paper, we understand some common trading behaviors by building networks of insiders in the form of graphs. These networks also help us in identifying if there are any anomalous trading behaviors associated with company insiders. Our analysis in this paper has resulted in some interesting patterns and scope for more future work in this area.

1. Introduction

In 2008 a company called SAC capital gained $275 million (profits made, and losses avoided), as result of one manager's insider trading, relating to two companies developing a drug. The manager then prompted several hedge funds to short more than $960 million of financial securities related to these companies. All this started when the manager of SAC capital was tipped off by a neurology professor who was part of the safety monitoring committee involved in the clinical trial. The information that the drug that was being developed has worse side effects was leaked before it was made public. This is by far the most significant insider trading case in the pharmaceutical industry. Such cases of insider trading are common both on a small-scale and large-scale. In some of these cases, the profits may be in thousands, but for some of them, the profits made by either a company, hedge fund or an individual are in millions. Detecting insider trading helps to avoid financial market abuse and stabilize the market.

Insider trading is of two types; first is legal insider trading. Officials of a company are allowed to buy and sell stocks of their company at any time following some rules. Insider trading becomes illegal when an official trade his/her company's stocks with prior knowledge of non-public information. In this paper, we are more concerned about the illegal insider trading and the people involved. Linking a person with insider trading does not mean he/she is an employee of that company. In many scenarios people at higher positions (CEO, CFO), with crucial nonpublic information like mergers between organizations or new product launch act as tippers to family and friends, who in turn take advantage of the information.

How to find insiders? SEC requires anyone who is buying or selling their company's stocks to fill Form-4, which contains information - on the number of stocks traded, the price at which they trade and remaining holdings after the trade. All these fillings are made public through the EDGAR system. In this paper, we analyze this data to find relations between insiders in a company. Over the past SEC merely waited for tips or Financial Industry Regulatory Authority(FINRA) referrals to find cases and investigate. In recent years SEC has taken strides to find and explore leads from billions of rows of data using data mining techniques.

As traditional analysis techniques are not sufficient to analyze such volumes of data, we try to build graphs for insiders to help quickly identify anomalous insider behavior. The graphs can also be studied to find unusual patterns of insider relations and their corresponding trading behaviors. We also explored additional ways of building networks of insiders to obtain higher order relationships between insiders. We have found some interesting results.

2. Related Work

The work of Kulkarni et al. [1] inspired us to work on this project. We reproduced most of the work done in this paper with minor modifications and additions to the approaches followed by the authors. They captured the relationship between insiders of each company by constructing graphs. They aimed at identifying interesting patterns which can indicate potential anomalies. They applied similarity based approach and LCS based approaches as done in [2] to their data. Then they explored additional ways to build networks of higher order relationships among the traders. They quantified the profits made by insiders as done in [2] by computing their signed normalized dollar amount.

For graph based anomaly detection, a variety of methods have been introduced out of which only a few were applied in the area of detection of illegal insider trading. The approach depends on the nature of the graph and a complete survey of this is done in [5]. An extensive large-scale analysis of insiders' trades using the Form 4 filings is done in [2]. They performed analysis using time series data mining where they discovered temporal patterns by partitioning the trades on several attributes. These attributes include company sectors, transaction types and corporate roles.

The authors of study [3] developed the OddBall algorithm in which OddBall refers to a sphere around each node. They focused on specific egonet patterns like star pattern, clique pattern etc which characterize the way ego node is connected to neighbors uniquely. They also showed that the egonets in networks of different domains obey some interesting patterns like Egonet Density Power Law, Egonet Weight Power Law, Egonet Rank Power Law, etc.

3. Data

There are many data sources like EDGAR, Insider Monkey, Google Finance etc. available online to mine for the insider trade data. We chose to use the existing MySQL database prepared from the data scraped from Insider Monkey and Google Finance. We got this data from our instructor and Teaching Assistant. This database contains data of over 1 Million trades, insider names, their positions, their trade type (purchase or sale), company information and opening and closing price of the stocks of each company. Table 1 below shows the summary statistics of this data.

Total Companies	12,485
Total Insiders	70,408
Total Sale Transactions	757,194
Total purchase Transactions	311,013

Table 1: Statistics

4. Anomaly detection by Pairwise comparison

Initially we processed the data to construct networks based on the trading trends of each pair of insiders belonging to a company. We constructed graphs where edges represented commonality of trade dates and nodes represented the insiders, and extracted connected components from the graphs to analyze for anomalies. We followed the approaches as done in [1].

4.1 Similarity based approach

In each company, for each unique pair of insiders, if they traded on at least 5 common dates (to filter insignificant shared behaviors), we computed the similarity score. We used a similarity function which takes the trade dates of insiders into account and proportion of dates that were common between an insider pair. We considered the insiders as nodes and added an edge between each pair if the similarity score S is > 0.5. This score ranges from 0 to 1. Following is the computation function for similarity measure:

$$S\,(X_C, Y_C) = \frac{(\sum_{i=1}^{|X_C|}\sum_{j=1}^{|Y_C|} I(x_i,y_j))^2}{(|X_C| \times |Y_C|)}$$

I() = 1, if $x_i = y_j$, 0 otherwise.

Table 2 below shows the network statistics for the purchase and sale networks.

Network	Nodes	Connected Components
Sale	1476	530
Purchase	1360	380

Table 2: Network Statistics (based on S)

Figure 1 and Figure 2 below show the distribution of connected components in purchase and sale networks.

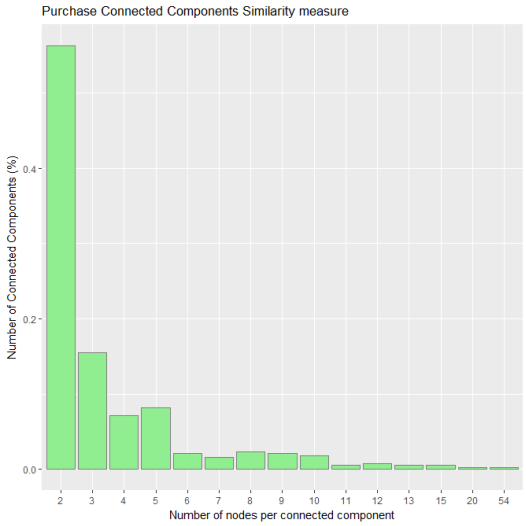


Figure 1: Distribution of connected components for purchase network (based on S)

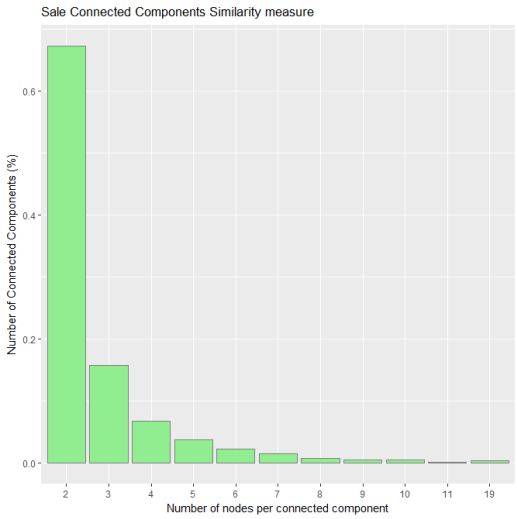


Figure 2: Distribution of connected components for sale network (based on S)

With this approach, temporal ordering of trading dates is not accounted for. So, we followed the Longest Common Subsequence based approach which is described below to account temporal ordering.

4.2 Longest Common Subsequence based approach

To account for the temporal ordering of trade dates, if a unique pair of insiders in a company shared a sub-sequence of dates of length at least t, we added an edge between them. By following this LCS-based approach, we constructed graphs and extracted connected components from them. We chose the threshold t based on the distribution of length of longest common subsequence between insider pairs in both purchase and sale networks. We considered insiders with t > 10 for purchase network and t > 5 for sale network. Table 3 below shows the statistics for purchase and sale networks.

Network	Nodes	Connected Components
Sale	3819	1099
Purchase	977	241

Table 3: Network Statistics (LCS-based)

Figure 3 and Figure 4 below show the distribution of longest common subsequences for purchase and sale networks.

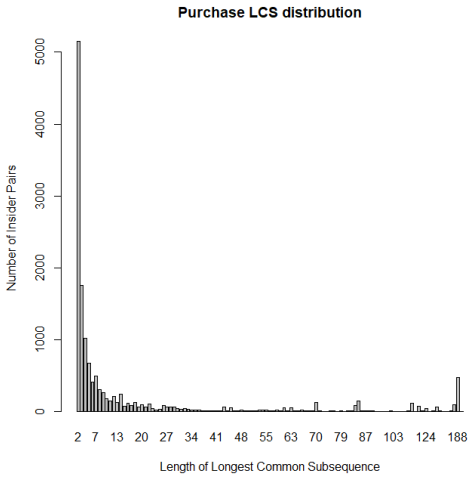


Figure 3: Distribution of Longest Common Sub-sequences of purchase network.

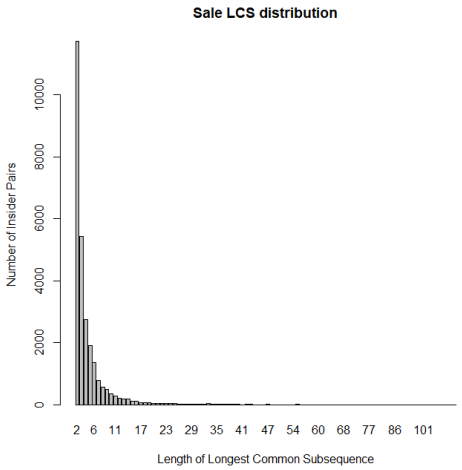


Figure 4: Distribution of Longest Common Sub-sequences of sale network.

Figure 5 and Figure 6 below show the distribution of LCS-based connected components for purchase and sale networks.

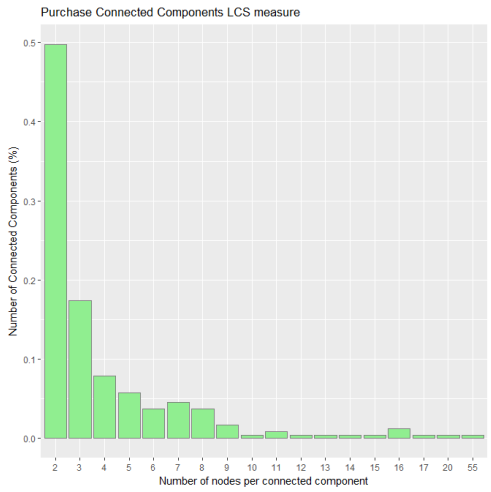


Figure 5: Distribution of Connected Components (LCS-based) – Purchase Network

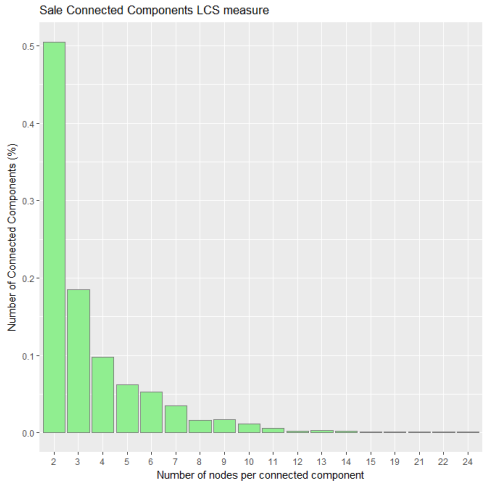


Figure 6: Distribution of Connected Components Components (LCS-based) – Sale Network

4.3 Connected components and egonets

We found out few interesting examples of connected components and egonets from the graphs constructed for purchase and sale networks using Similarity and LCS-based approaches.

4.3.1 Similarity based approach

Figures 7(a) and 7(b) below show the examples of connected components constructed using similarity measure S for purchase and sale networks. These connected components have very high connectivity among the insider nodes which indicates frequent pairwise similarities between the insider pairs.

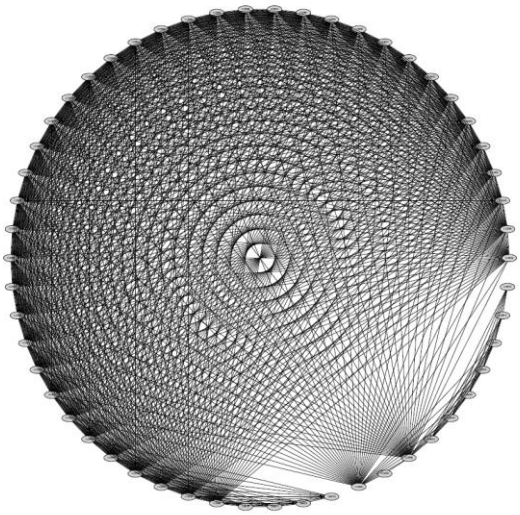


Figure 7(a): Purchase: Connected Component of International Speedway Corp Class A and Class B

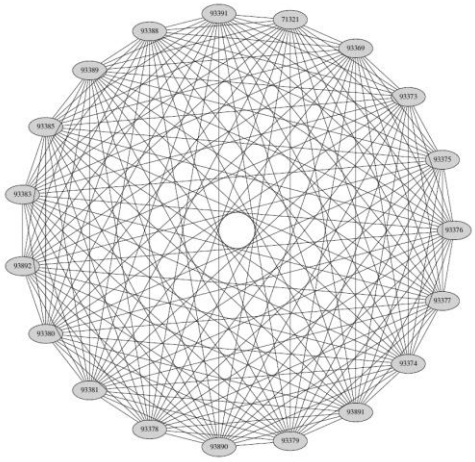
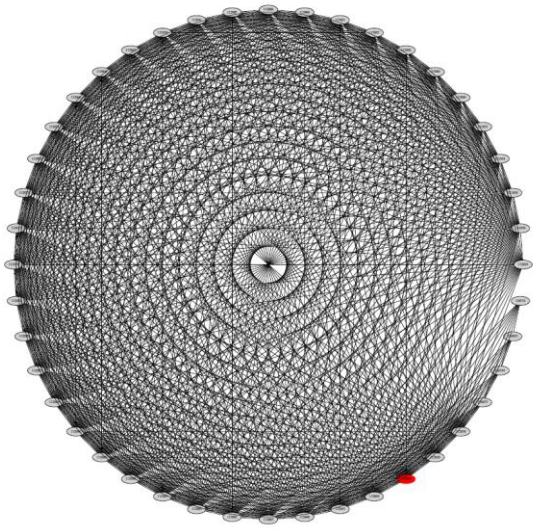
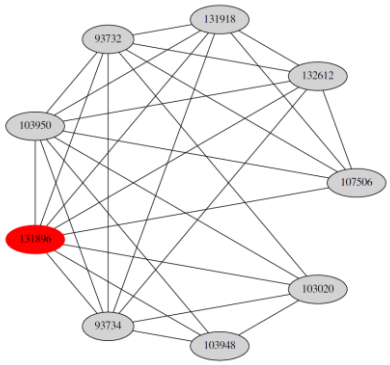


Figure 7(b): Sale: Connected Component of Vantiv. Inc

Figures 8(a), 8(b),8(c) and 8(d) below show the examples of discovered anomalous egonets for purchase and sale networks with higher total outlier scores. These anomalous ego nodes indicate the role of hubs between cliques. These often occupy a bridge position between the connected components.

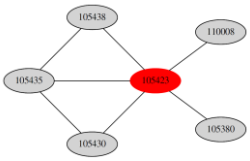


International Speedway Corp



Company 34947 (Not given in the data set)

Figure 8(a) and 8(b): Purchase: Egonet of insiders with highest outlier scores.



WINN-Dixie Stores, Inc

WCRX

Figure 8(c) and 8(d): Sale: Egonet of insiders with highest outlier scores.

4.3.2 LCS-based approach

Figures 9(a) and 9(b) below show the examples of connected components constructed using LCS-based approach for purchase and sale networks. These connected components have very high connectivity among the insider nodes which indicates many common subsequences greater than the threshold between the insider pairs.

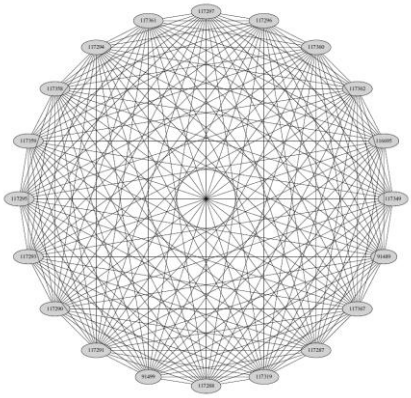


Figure 9(a): Purchase (LCS-based): Connected Component of Hyster-Yale Materials Handling Inc.

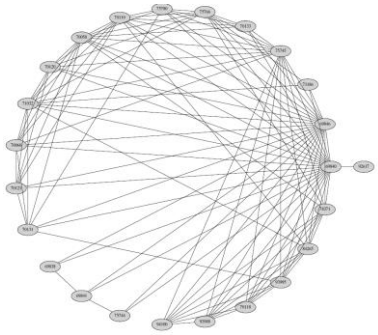


Figure 9(b): Sale (LCS-based): Connected Component of General American Investors

Figures 10(a), 10(b) and 10(c) below show the examples of discovered anomalous egonets for purchase and sale networks with higher total outlier scores. Similar to what we observed in Section 4.3.1, These anomalous ego nodes also behave as hubs between cliques.

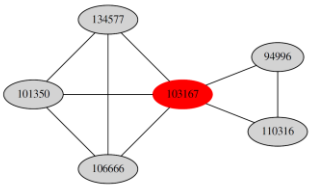


Figure 10(a): Purchase (LCS-based): Egonet of insider with highest outlier score, First Mid-Illinois Bancshares

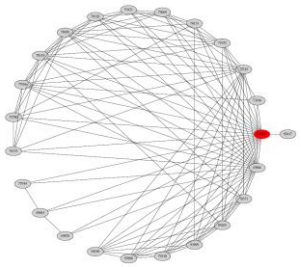


Figure 10(b): Sale (LCS-based): Egonet of insider with highest outlier score of General American Investors

4.4 Power Law Fitting

After computing the egonets from the graphs produced by similarity based approach and LCS-based approach, they are analyzed for discovering anomalies. For an ego net corresponding to ego node u, we computed number of nodes V_u and number of edges E_u. We plotted V_u against E_u for all the egonets across all companies. This plot revealed a power law relationship which we used to define anomalies. First, we computed the least squares line fit on the values of $\log(E_u)$ and $\log(V_u)$ and then used the fit parameters to get the outlier scores for each ego node. Here, the objective was to measure the deviation of each node u from the power law as done in [2] and [3].

Following is the computation function of outlier score:

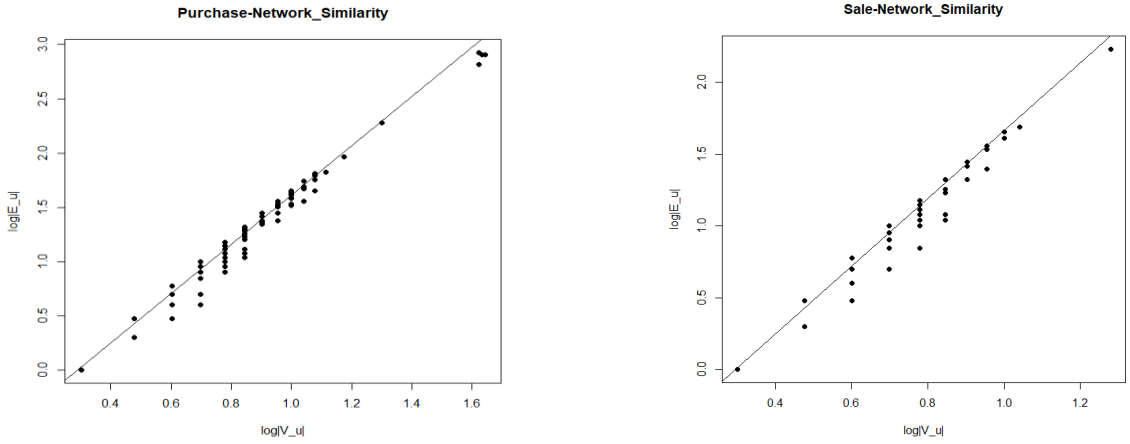
$$\text{Score}\ (u) = \frac{\max(E_u, f(V_u))}{\min(E_u, f(V_u))} \times (\log(|E_u - f(V_u)| + 1))$$

Then, we computed the Local Outlier Factor (LOF) for each ego node as done in [4]. LOF measures the ratio of density of ego node u to the average density of its k nearest neighbors. We set the value of k to 5. We added LOF

score to Score (u) to obtain the Total Outlier Score as done in [3]. The rationale for choosing LOF is to complement the distance from line based score with a density based score. Otherwise, a point far from all other points but still on the power law fit will not be considered outlier.

$$TotalOutlierScore(u) = Score(u) + LOF(u)$$

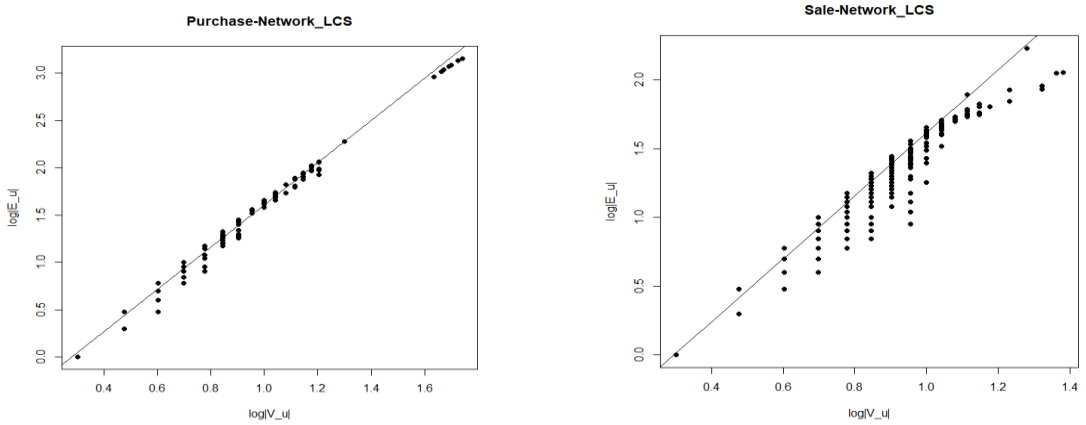
Figures 11(a) and 11(b) below show the Power Law Fitting plots for Purchase and sale networks constructed using Similarity measure. The points that are highly deviating from the fit line have higher outlier scores.



(a) (b)

Figure 11: Power Law Fitting (Similarity based): (a) purchase, (b) sale

Figures 12(a) and 12(b) below show the Power Law Fitting plots for Purchase and sale networks constructed using LCS-based approach. The points that are highly deviating from the fit line have higher outlier scores.



(a) (b)

Figure 12: Power Law Fitting (LCS-based): (a) purchase and (b) sale

5. Evaluation with signed normalized dollar amount

Till now we found interesting cases with the above two approaches. Now, we tried to quantify the above results by looking at the profit, the identified traders made during the sequence of dates shared with another trader. For this we computed the signed normalized dollar amount as described in [2] of insiders obtained by LCS-based approach. Also, we computed the signed normalized dollar amount of every insider of every company to see if we can find any

interesting anomalies. An insider makes profit when he/she buys stocks at a price lower than the closing price or when he/she sells stocks higher than the closing price of the stock on that day. So, the signed normalized dollar amount ranges between -1 and 1. Profits range from 0 to 1 and losses range from 0 to -1. Following is the function to compute signed normalized dollar amount R of each insider.

$$R = \frac{Transaction\ Price\ X\ \sum Shares\ Traded}{Dollar\ Volume}$$

Where, Dollar Volume DV = Total number of shares traded on a day x market closing price of the stock on that day.

5.1 LCS-based

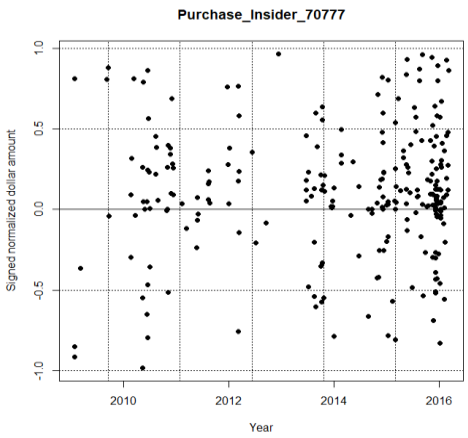
Figures 13(a), 13(b), 13(c) and 13(d) below show the plots of time series of signed normalized dollar amounts of LCS-based ego nodes of purchase and sale networks. Since the trade prices are not always available in the data, we could represent only a subset of the total transactions. For the following figures we observe that majority of the transactions are indicated above 0 level which indicates repeated profit. 13(a) shows transactions of an insider who traded stocks of multiple companies and made number of profits especially in the year 2016. This insider's total outlier score is 1.054004865 which ranked 70. There is only one entry with LCS score = 21. So, we got an egonet for him connected to only one other insider.

Below screenshot shows the trades of insider 70777 with multiple companies:

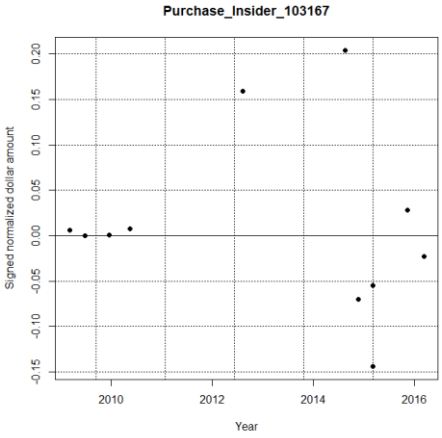
	A	B	C	D
	company_stockID	insider1_id	insider2_id	lcs_score
	32841	109302	70777	1
	32841	132812	70777	2
	32841	125320	70777	2
7	35522	127554	70777	1
7	36026	101349	70777	1
6	36026	103462	70777	1
0	36026	126499	70777	1
7	36026	126513	70777	1
1	36026	86514	70777	1
2	36572	89350	70777	2
8	37069	138627	70777	2
1	37069	103462	70777	3
2	5511	119573	70777	1
0	5511	119577	70777	1
1	5511	119575	70777	1
3	5674	103620	70777	3
5	5723	126404	70777	1
1	5934	123809	70777	1
0	6159	119811	70777	1

	company_stockID	insider1_id	insider2_id	lcs_score
	32853	70777	88782	3
0	36026	70777	126509	1
1	36026	70777	126492	1
1	36026	70777	126507	1
6	36026	70777	119481	1
1	4230	70777	95087	2
5	37069	70777	114367	1
6	5674	70777	117547	2
4	39127	70777	95087	21

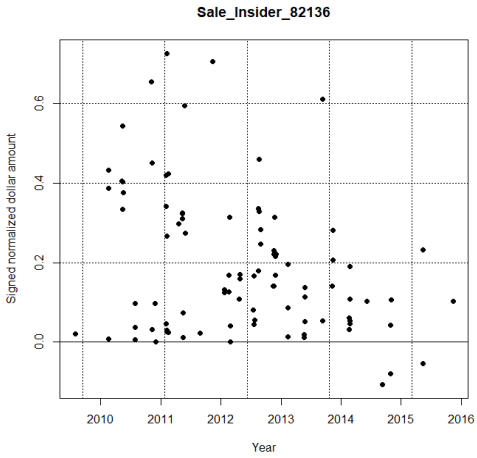
Figure 13(b) shows the insider with highest outlier score but few trades. Figure 13(c) shows the insider who made only profits 95% of the time which is suspicious and hence a potential anomaly. This insider's total outlier score is 1.013247896 and ranked least of all insiders. Also, Figure 13(d) shows 1 transaction that gave huge profit to the insider.

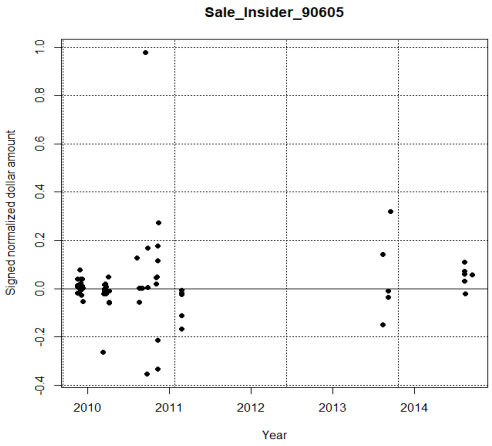


(a)



(b)





(c) (d)

Figure 13: Time series of Signed Normalized Dollar Amount (LCS-based) egonets (a) and (b) Purchase network. (c) and (d) Sale network.

5.2 Every Insider

We also plotted the time series of signed normalized dollar amount of every insider and found interesting cases which were not found by LCS-based approach. We considered all ratios, R above 0.09 and looked for interesting scenarios. Figure 14(a) and 14(b) below show examples of Signed Normalized Dollar Amounts of insiders in purchase and sale networks respectively. In Figure 14(a), the insider traded only on few dates of the year 2011 and 2012 Jan. We could observe that he made huge profit in Jan and considerable amounts of profits in the remaining dates. In Figure 14(b), the insider made profits on most of the dates. In the year 2012 and 2015, he made huge profits.

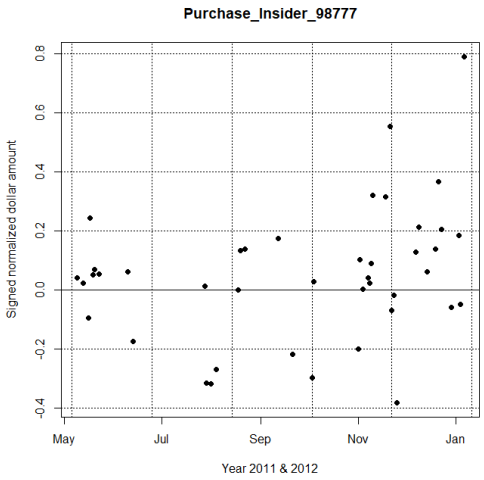


Figure 14(a): Time series of Signed Normalized Dollar Amount of an insider (Everyone-based) in Purchase network

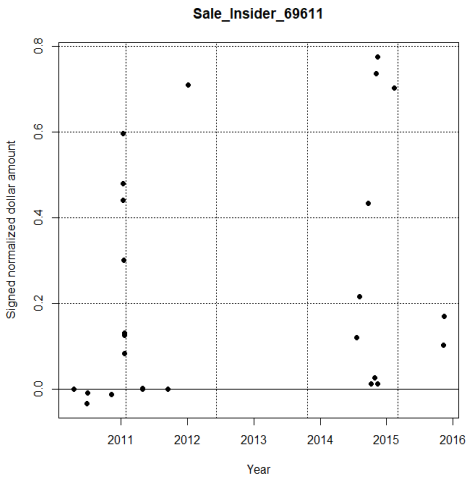
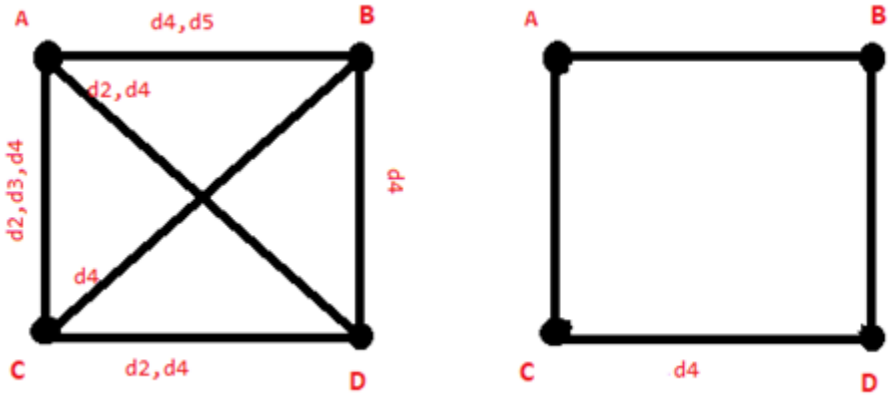


Figure 14(b): Time series of Signed Normalized Dollar Amount of an insider (Everyone-based) in Sale network

6. Hypergraphs

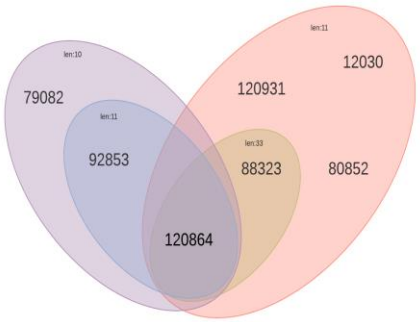
Graphs are helpful to build networks of insiders and find interesting patterns, but they have a limitation of showing only pairwise relations. So, to capture multiway co-occurrence between insiders we use hypergraphs. A hypergraph

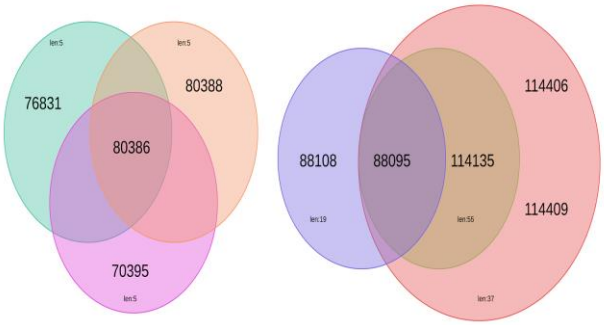
is a generalization of a graph in which an edge can join any number of vertices. It is represented as a pair H = (V, E), where V is a set of elements called nodes or vertices(Insiders) and E is a set of non-empty subsets of V called hyperedges or edges. Let us consider four insiders A, B, C and D and their corresponding sequence of trade dates A[d1,d2,d3,d4,d5], B[d4,d5], C[d2,d3,d4,], D[d2,d4]. Below are the representations of these four insiders based on the dates they traded.



a) Graph representing A, B, C and D b) Hypergraph representing A, B, C, and D

In the above figure a represents dates shared by pairs of insiders like A and C share dates sequence of d2, d3, and d4 while insiders C and B share only d4. Figure b shows the four-way cooccurrence where all the insiders share the common date d4.
We have presented below two hypergraphs generated from our data. The hypergraphs are constructed based on the LCS of the insiders. The thresholds of 10 and 5 are taken into consideration for purchase and sale respectively when constructing hypergraphs.





Hypergraph for Ticker-QNBC for purchase Hypergraph for Ticker-CUK for sale

By looking at the above hypergraphs inspecting further about the insiders obtained by the intersection of multiple hyperedges should give some interesting results or anomalous entity. The hyperedges are annotated with the length of the LCS they share. We notice that from sets of two to n as the number of insiders in sets increases the length of the LCS decreases significantly.

7. Conclusion

In this project, to capture insider trading behaviors and their relationship to other insiders, we analyzed the insider trading data collected from Insider Monkey and Google Finance. We constructed different kinds of graphs and analyzed for anomalies. We discovered interesting patterns of traders while evaluating the results. Few examples of these are discussed in the above sections. More work can be done to explore the complex patterns captured by the hypergraphs approach.

8. Acknowledgement

We thank Priya Mani and Dr. Domeniconi for providing us the data set and for answering our many questions.

9. References

[1] Kulkarni, Adarsh, Priya Mani, and Carlotta Domeniconi. "Network-based Anomaly Detection for Insider Trading." arXiv preprint arXiv:1702.05809 (2017).

[2] Tamersoy, Acar, et al. "Large-scale insider trading analysis: patterns and discoveries." Social Network Analysis and Mining4.1 (2014): 201.

[3] Akoglu, Leman, Mary McGlohon, and Christos Faloutsos. "Oddball: Spotting anomalies in weighted graphs." Advances in Knowledge Discovery and Data Mining (2010): 410-421.

[4] Breunig, Markus M., et al. "LOF: identifying density-based local outliers." ACM sigmod record. Vol. 29. No. 2. ACM, 2000.

[5] Akoglu, Leman, Hanghang Tong, and Danai Koutra. "Graph based anomaly detection and description: a survey." Data Mining and Knowledge Discovery 29.3 (2015): 626-688.

[6] https://www.lexology.com/library/detail.aspx?g=76c13e2a-8097-40ce-98b9-e9ba0a874061
[7] http://www.geeksforgeeks.org/printing-longest-common-subsequence/

10. Contribution of each team member

Contribution of Sameera Bammidi:
Contribution towards writing the paper:
I am responsible for and wrote sections 2, 3, 4 ,4.1 ,4.2 ,4.3 ,4.3.1 ,4.3.2, 4.4, 5, 5.1, 5.2, 7

Contribution towards project:
Processing the data:
Downloaded and installed MySQL to view and query for data tables:
https://dev.mysql.com/downloads/installer/
Referred to the following tutorial:
http://theopentutorials.com/tutorials/java/jdbc/jdbc-mysql-connection-tutorial/

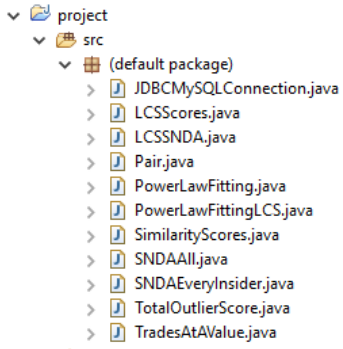
Used java with Eclipse IDE to compute Similarity scores, LCS scores, filtering with thresholds, construct graphs, connected components, egonets and produce .csv files required for power law fitting and constructing graphs, connected components and egonets.

I used the algorithm for computing LCS from the following reference:
https://www.programcreek.com/2014/04/longest-common-subsequence-java/

I used mysql-connector-java-5.1.41-bin.jar to connect to MySQL database and jgrapht-core-1.0.1.jar to construct graphs, find connectivity and construct connected components and egonets.

Following are the references I used while importing the jar files and writing the graph related code:
http://jgrapht.org/
http://jgrapht.org/javadoc/org/jgrapht/alg/ConnectivityInspector.html
http://graphstream-project.org/doc/Tutorials/Working-with-algorithms-and-generators/
https://code.snipcademy.com/tutorials/data-structures/graphs/adjacency-matrices

Following is the screenshot of all the 11 java class files I wrote:

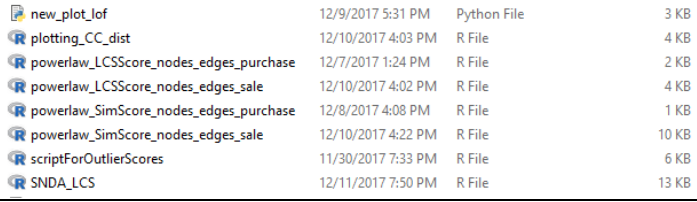


Used Python to compute LOF after obtaining the Outlier scores. Below are Python code references:
http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html
https://matplotlib.org/api/pyplot_api.html
http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Evaluation:
Used Graphviz to visualize graphs, connected components and egonets:
https://graphviz.gitlab.io/download/
http://www.webgraphviz.com/

Computed histograms of connected components, computed Outlier scores, computed and visualized the power law fitting and visualized signed normalized dollar amount using R.
Following is the reference I used:
https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html

Following is the screen shot of the 7 R script files and 1 python file, I wrote:



Contribution of Divya Vajja

Contribution towards writing the paper:
Abstract, Sections 1, 6

Contribution towards project:
Analyzed the data and created hypergraphs for purchase and sale for the companies that seemed interesting. Hypergraphs were constructed using java programming language. Visualized two hypergraphs manually.
References used:
http://jung.sourceforge.net/doc/index.html
http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/graph/SetHypergraph.html
https://www.lexology.com/library/detail.aspx?g=76c13e2a-8097-40ce-98b9-e9ba0a874061
http://www.geeksforgeeks.org/printing-longest-common-subsequence/