UPPSALA
UNIVERSITET

PROJECT REPORT

Stock market networks & Information events

Analysis of the relationship between an Empirical Investor Network and information events in the stock market

Andreas Kieri & Joakim Saltin
Supervisor: Johan Walden, assistant professor of finance at University of California at Berkeley

Project in Computational Science: Report

January 2012

Department of Information Technology

Abstract

Information flow between traders at a stock market is known to have a strong impact on trading behavior. By viewing the market as a network of traders, we investigate the hypothesis that traders who are centrally located in the network receive valuable information earlier than non-central traders, and because of this will be able to make better investment decisions. We do this by implementing an Empirical Investor Network (EIN) based on real transaction data from the Istanbul Stock Exchange. We extend previous research by refining algorithmic computations in order to better simulate real life information diffusion. Furthermore, we examine if centrally located traders act earlier than non-central traders in relation to specific information events, by investigating trading behavior in the days surrounding published news stories. We find that central traders indeed act earlier and that they also generate bigger profits, both throughout the year and in relation to specific information events.

Keywords: *Information events, stock market networks, centrality*

Contents

1 Introduction 4

2 Theory 5
2.1 Model: Investor Network 5
2.2 Empirical Investor Network 6
2.2.1 Testable properties of the stock market using the EIN 6
2.3 Network measures 6
2.3.1 Connectivity matrix 6
2.3.2 Centrality measures 7
2.3.3 Power iteration 8
2.4 Profits 9

3 Implementation 9
3.1 The data 9
3.2 Algorithm for creating the connectivity matrix 10
3.3 Limitations 11

4 Project tasks and results 12
4.1 No brokerage house connections 12
4.2 Splitting the data sample 13
4.3 Twenty-four-hour connection window 14
4.4 Information Events 17

5 Conclusions 20

6 Acknowledgements 21

7 References 21

8 Appendix: Event details 22

1 Introduction

Numerous studies have shown that communication is a key factor when traders on a stock market make their investment decisions. Based on this, a straightforward hypothesis is that traders who gain access to information earlier would be better informed and therefore be able to make better investment decisions than less informed traders. To verify this, however, is not an easy task. This project is part of a bigger research project[1] which aims to do just that, using real transaction data from the Istanbul Stock Exchange (ISE) during the year of 2005.

The approach of the research, developed by H. Ozsoylev, J. Walden et al., is to model the stock market as a network of traders and then approximate this network by constructing a proxy – an Empirical Investor Network. This is done by extracting possible information links from the transaction data based on the assumption that connected traders will trade in a similar way. If one can show that traders who are well-connected (i.e. central in the network) both make bigger profits and trade earlier than less connected traders, then this would support the view that information is diffusing through the network and that well-informed traders use this to their advantage.

Our project mainly focuses on verifying that central traders act early. This is done by analyzing trading behavior of all traders on the ISE in the year of 2005 in relation to a number of large stock movements which were related to information events, e.g. spreading of rumors. These results are then regressed on the computed centralities of the traders, with the aim of finding a positive correlation between early trading and centrality. The project also included extending previous analyzes of the centrality measures, e.g. by increasing the time horizons and performing out-of-sample tests, in order to further verify the robustness of the existing results, as given by the work of students in a previous instance of this course[2]. Our project also builds upon their code.

[1] H. Ozsoylev, J. Walden et al.: "*Investor Networks in the Stock Market*"

[2] N. Ericson, L. Larruy.: "*Network analysis of a stock market*"

2 Theory

2.1 Model: Investor Network

The network model introduced by Ozsoylev, Walden et al. (2012) is a way to model a stock market, incorporating the important topic of how the flow of information affects the performance of individual investors. The motivation for this model is that there is extensive evidence that information diffusion plays an important role in investment decisions. [3]

The model assumes a certain number of traders N_I that are in some way connected, and also a large number of uninformed traders N_U, whose trading motives are not modeled.

The trades are thought of as occurring at discrete times, with discrete time intervals between trades. At each point in time, one of the informed traders receives a valuable piece of information, based on which the trader then trades (e.g. buy, sell, or short sell one or more stocks) with an expected profit π that is positive and above expected market returns. The opposite side of this trade, which results in a loss $-\pi$, is assumed to be taken by one of the uninformed traders. To model the diffusion of information through the network, there is a certain probability that the trader shares the information with one of his neighbors in the network during the next time step. The receiver of the passed on information then trades on it, with an expected profit that is positive, but less than the previous trader's profit. In practice, this occurs because as time passes, information is incorporated into the prices due to the previous traders trade (or slow diffusion of information through other channels).

Consider for instance the network in figure 1. The information diffusion could work like this: trader 1 receives and trades on a valuable signal at time t_0, and then shares the information with trader 2, who in turn trades on it at time t_1. The diffusion is then completed at time t_2 when trader 5 has received the information from trader 2 and trades on it, after which the signal has been fully incorporated into the stock price and there can be no more profits made on that specific signal.

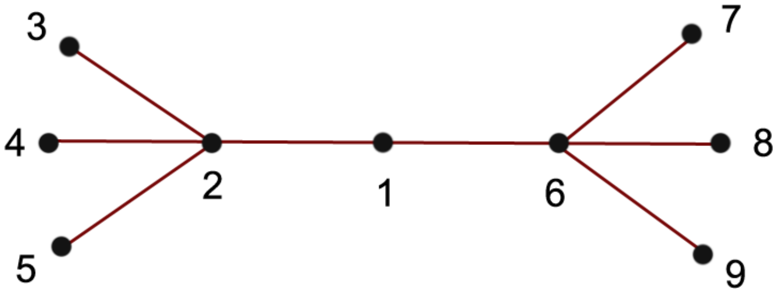


Figure 1: A network with nine traders

[3] Shiller, R. J., and J. Pound (1989): *Survey evidence on the diffusion of interest and information among investors*, Journal of Economic Behavior, 12, pp.4766

An intuitive implication of this model is that the more *well-connected* you are (see centrality in section 2.3.2), the more profitable you will be in your trading since you will have a greater chance of receiving these valuable signals earlier, before they become incorporated into the stock prices. Ozsoylev, Walden et al. show this to be accurate by simulating trades in a theoretical investor network consisting of 50 investors. [4]

2.2 Empirical Investor Network

In a real application, the actual connections in the network are of course impossible to establish properly, given the vast amount of traders in the market. Instead, a proxy of the true network, called an Empirical Investor Network (EIN), is constructed based on the idea that connected traders will tend to have a similar trading pattern. This is done by assuming that connected traders will (from time to time, at least) trade in the same stock in the same direction – both buying or selling – within a certain period of time, a *connection window*, of each other. By this assumption, the structure of a network in any given stock market can be extracted from transaction data (see section 3.2 for details).

Even though an EIN constructed in this way will inevitably contain false connections (people who randomly trade in the same stock without being connected), it has been shown that the EIN is usually a good proxy of the true underlying network. This has been shown using a Monte Carlo simulation on a theoretical network of 100 traders, where the fraction of false connections ended up being just 10% [5].

Furthermore, a previous study on real stock market data show that the identified EIN is quite stable over time [6], which is a requirement for a good EIN, since the true network is unlikely to change much over time.

2.2.1 Testable properties of the stock market using the EIN

The main ideas that the investor network model is based on is that traders who are central in the network get hold of good information quickly, and thus both trade earlier and make bigger profits than less central traders. The introduction of the EIN allows for these properties to be tested against real transaction data. Based on the centrality calculated using the EIN, one can relate each individual trader's centrality to his profits, and check if there is indeed a correlation between the two. Another test is to study large market movements that were caused by information diffusion (e.g. rumors), and see if there is a correlation between being central in the network and trading early.

2.3 Network measures

2.3.1 Connectivity matrix

In order to analyze the network and extract useful information, the connections between actors in the network need to be established. Once this is done, the connections can be stored in a *connectivity matrix* (also commonly referred to as an *adjacency graph*) M, where each element M_{ij} is either zero or one,

[4] H. Ozsoylev, J. Walden et al., 2012: "*Investor Networks in the Stock Market*", pp.3-10

[5] H. Ozsoylev, J. Walden et al., 2012: "*Investor Networks in the Stock Market*", pp.10-12

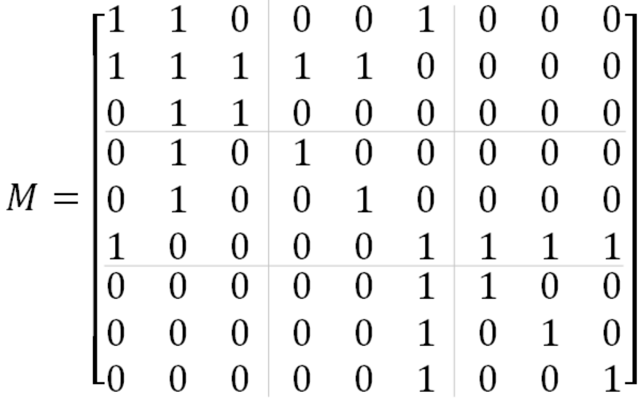
[6] N. Ericson, L. Larruy., 2010: "*Network analysis of a stock market*", p.13

representing a connection from actor i to actor j. In our investigations, we have assumed that the connections are bidirectional, meaning that $M_{ij} = M_{ji}$ which leads to a symmetric connectivity matrix. Furthermore, we have for technical reasons considered traders as self-connected, i.e. all elements $M_{ii} = 1$. In total, the resulting connectivity matrix will be:

$$M^{total} = \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}$$

where M is N_I by N_I and I is an identity matrix of rank N_U.

As an example, consider again the network in figure 1, which shows the informed traders and their connections. Assuming bidirectional connections and self-connectivity, the resulting connectivity matrix (ignoring noise traders) is as shown in figure 2. Inspection shows that the matrix is indeed symmetric, and that the traders are self-connected, since the main diagonal contains only ones. For instance, row (or column, equivalently, due to symmetry) one contains non-zero elements in column one, two and six. This means that trader one is connected to himself, trader 2 and trader 6, which is in accordance to figure 1.



$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Connectivity matrix corresponding to the network in figure 1

2.3.2 Centrality measures

Once the connections have been established, some interesting characteristics about the network can be extracted. Perhaps the most important one is a measure of how *central* a certain actor is in the network. There are a lot of different measures of centrality, but they all estimate the same thing: how well-connected each actor is.

The most straight-forward measure is the degree one D^1 (or simply *degree D*), which is just a count of how many direct neighbors an actor has. Returning to the example in figure 1, trader one has a degree of three, while trader two has a degree of five (including self-connectivity). The vector D, containing the degree of each agent, can be easily found from the connectivity matrix by summing each row (or column in the case of bidirectional connections) i:

$$D_i = \sum_j M_{ij} \tag{1}$$

What the degree one measure does not take into account however, is that the degree of the neighbors should also affect centrality. The intuitive reason for this is that a well-connected neighbor should be "worth more" than a poorly connected neighbor. To account for this, the degree n measure D^n is used, which takes into account the degree of all the neighbors within a connectivity distance of n from the actor. For instance, with $n = 2$, the degree two counts the actors neighbors and also the neighbors neighbors. Degree n is thus a recursive count of the degree of the neighbors that are within connection distance n. It follows directly that the degree n measure is a lot more expensive to compute than the simple degree measure.

A cheaper but equally informative measure is the eigenvector centrality C. It assigns centrality scores that are proportional to the centralities of the neighbors (multiplied by a scaling factor $1/\lambda$) according to:

$$C_i = \frac{1}{\lambda} \sum_j M_{ij} C_j \tag{2}$$

or, in vector form:

$$\lambda C = MC \tag{3}$$

This has the form of an eigenvalue problem, where C is an eigenvector of the connectivity matrix M. It can be shown that for the centralities to be positive, the proportionality constant λ must be the largest eigenvalue of M.[7] Since M is usually sparse in practice, the computation of eigenvector centrality can be done quite inexpensively using an iterative method.

In the empirical network investigated later in this report, noise traders who trade with high frequencies (and for other motives than basing trades on information) will appear as highly connected, which will influence the centrality measure. To compensate for this, a more robust measure is introduced; the *rescaled centrality* $\frac{C}{D}$, i.e. eigenvector centrality divided with the degree. High-frequent noise traders will thus be punished because of their high degree, leading to a lower rescaled centrality than that of a trader who is central, but has fewer direct connections (lower degree).

2.3.3 Power iteration

Power iteration is an iterative method for solving eigenvalue problems that is particularly suited to problems with sparse matrices. It is defined as follows: Start with a start-guess x_0. In each iteration k, calculate

$$x_{k+1} = \frac{Mx_k}{\|Mx_k\|} \tag{4}$$

until a convergence criterion on x_{k+1} is met. It can be shown that if the matrix M has a unique largest eigenvalue, then the sequence $\{x_k\}_{k=1,2,3,...}$ will converge into the eigenvector corresponding to the largest eigenvalue. Since each step only involves a matrix multiplication with a sparse matrix, the computational work is quite inexpensive. Also, convergence is usually achieved quite fast (in less than 50 iterations). [8]

[7] M. E. J. Newman: "*The mathematics of networks*", p.5

[8] H. Ozsoylev, J. Walden et al., 2012: "*Investor Networks in the Stock Market*", p.8

2.4 Profits

Ultimately, the performance of each investor is determined by their profits. The actual profits are given by the change in value of individual traders' portfolios (and the amount of dividend payments). Since there is no way to know the actual portfolio holdings of individual traders, we calculate approximations of the profits based on the change in value of the stocks between the time they are bought or sold and a later point in time, determined by a *profit window.* With this definition of profits, the mean profits among all the traders in the market must be zero, since there for every deal are two parties, one taking up the profit and the other one the loss. A profit will thus be to buy a stock that then grows in value or selling a stock that then decreases in value ("buying low, selling high"), while a loss will be incurred by buying a stock that increase in value or selling before a value increase ("buying high, selling low").

The motivation for why this approximation is reasonable is the standard assumption about stock markets that future returns are not predictable. This means that, on average, the time delay between the transaction day and the day when the profit is calculated will not infer a bias to the stock value, and the calculation should thus be a reasonable measure of the profitability of the decision to purchase/sell.

When analyzing trader performance, simply using the (absolute) profits will not be a good indicator, since it is likely to be influenced by trading volume. Instead, we use a *normalized profits* measure μ defined as a trader's profits P in Turkish Lira (TL) divided by the trader's trading volume V (also in TL):

$$\mu = \frac{P}{V} \tag{5}$$

This measure will include the overall market trends (e.g. price increases in the whole market), so another measure which further emphasizes an individual trader's performance is the *normalized excess profits* measure μ^e:

$$\mu^e = \frac{P^e}{V} \tag{6}$$

The excess profits P^e is found by subtracting the increase of the ISE index (which contains all stocks on the ISE) from the profits P. An excess profit $\mu^e = 0$ will thus mean that the trader has made profits corresponding to the market average, i.e. matching the index.

3 Implementation

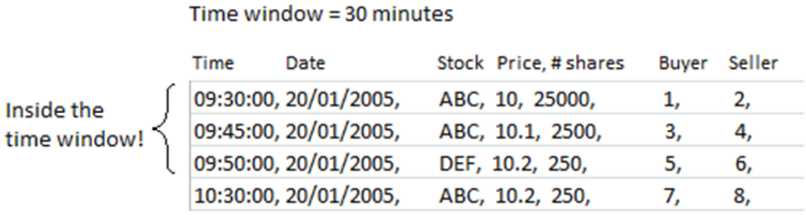
3.1 The data

Recall that we have no way of knowing anything about actual connections between traders, and therefore use the generated EIN as a proxy for the real information network. Thus, the connectivity matrix described in earlier sections will have to be constructed entirely by analyzing raw transaction data from an actual stock market. For this purpose we were given data from the Istanbul Stock Exchange (ISE), which was founded in 1986 and is the only corporation in Turkey providing trading in securities. The data set consists of all transactions involving stocks, performed at the ISE during the year 2005. At the ISE, traders enter their buy and sell orders electronically which are then matched by a computer system. Over the year, shares in 303 different stocks were traded and about 580,000 traders were active. In the raw data, each trader is represented by a trader account that belongs to a brokerage house, and is classified as either private or institutional. Traders are uniquely identified by combining their account number and their brokerage house, which enables us to represent each trader with a unique local trader number. We make no difference between institutional and private traders. During the year 2005, over 43 million stock transactions were performed at the ISE. The raw data is contained in one single file where each line holds information concerning a specific transaction. For each trade the following pieces of information are available: date and time of the trade, stock ID, number of shares traded, price per share, the account number and broker of the buyer, the account number and broker of the seller, and whether the trade was a short sell. Furthermore, stock splits and dividends will influence the profit calculations. These are taken into account by using lists containing all splits and dividends for each stock over the year.

3.2 Algorithm for creating the connectivity matrix

The aim is to extract a connectivity matrix from the raw transaction data. To do this, a condition for two traders to be considered as connected has to be established. Since it is assumed that traders act upon the information they receive; two traders that act in the same way might have access to the same pieces of information, and might even have shared this information with each other. With this in mind a reasonable condition for two traders to be connected would be that they have traded in the same way, i.e. both traders have either bought or sold the same stock within a specified connection window, ΔT, from each other. For instance, a trader will probably sell a certain stock if he/she has information that indicates that the stock price will drop in the near future. If this information is shared with another trader, he/she will most likely act in the same way and also sell the stock. If the second trader also sells the stock within ΔT from the first one, we register a connection between the two traders. Thus, to construct the connectivity matrix, situations like this have to be identified in the raw data. This is done by comparing each transaction with all other transactions performed within one connection window. Note that traders who are on opposite sides of a trade, i.e. the buyer and the seller in a transaction, are not considered as connected since they act in opposite ways. Figure 3 shows a segment of the transaction data, although greatly simplified, to demonstrate a situation where connections would be registered.

In this example, where the first transaction is being compared to the others and a connection window of 30 minutes have been used, traders 1 and 3 would be registered as connected since they bought the same stock (ABC) within ΔT. The same goes for trader 2 and 4, who both sold the same stock within the connection window. Note that no connections would be registered for trader 5 and 6 since they didnt trade in the same stock as any of the others, although inside the connection window. Also, no connections would be registered for trader 7 and 8 since they traded outside of the connection window compared to the first transaction.


Time window = 30 minutes

Time	Date	Stock	Price, # shares		Buyer	Seller
09:30:00,	20/01/2005,	ABC,	10,	25000,	1,	2,
09:45:00,	20/01/2005,	ABC,	10.1,	2500,	3,	4,
09:50:00,	20/01/2005,	DEF,	10.2,	250,	5,	6,
10:30:00,	20/01/2005,	ABC,	10.2,	250,	7,	8,

Inside the time window!

Figure 3: Transaction data example

The Deque – double ended queue
Clearly, comparing each line with all other lines in the data would be very inefficient, since a connection between two traders only can be registered if the corresponding transactions were performed within one connection window. Hence, a method in which we only compare transactions that will potentially give rise to a connection is needed. For that reason, a deque - a double ended queue is being used. A deque is an array of data lines where transactions enter at the bottom and are pushed out at the top. At any moment it will contain all transactions performed within one connection window compared to the transaction at the top. When the transaction at the top has been compared with all others in the deque, it is pushed out and new lines are entered at the bottom until the deque once again holds all transactions within one connection window, compared to the one at the top. In this way the deque ”works” its way through the data set until all transactions have been checked for connections.

Threshold
A larger connection window will obviously capture more connections resulting in a lot of noise. That is, traders that are not actually linked become connected in our model simply because they independently happen to trade in the same way. Noise connections will inevitably arise but the number of them can be greatly reduced by using a threshold, M. The more times a connection between two traders can be registered, the higher the probability is that they are indeed sharing information with each other. When using a threshold, all connections below the threshold value are removed, only keeping connections between traders that have traded in the same way over and over again. The threshold will also be essential when dealing with memory issues that arise when the connectivity matrix becomes too dense.

3.3 Limitations

All the stock transactions performed at a stock market during a year is a huge data set from which information is to be extracted. As with all large datasets this imposes some difficulties in terms of calculations. Therefore the computations are executed on the Uppmax System at Uppsala University. More specifically the Kalkyl cluster at Uppmax, which consists of 348 compute servers with varying amounts of memory, will be used. Some of the nodes have a 72 GB RAM and these are the ones being used for our calculations since the whole connectivity matrix must fit into memory. Although Uppmax enables high performance computations there are still problems that occur when the connection window ΔT becomes larger. These problems concern both memory and computational time. The memory will become a problem as the connectivity matrix becomes denser, resulting from more connections being registered, until it no longer fits in memory. The computational time problem is related to the fact that each line in the raw data has to be compared with all other lines in the deque. As the connection window grows, the number of line comparisons will grow exponentially. The computational time will also be affected when the connectivity matrix exceeds memory. When this happens the computer has to perform page swaps when accessing elements and all operations involving the matrix will become very slow. Fortunately, as can be seen in the section 4.3, there are ways to overcome these problems.

4 Project tasks and results

The tasks in this project can be broken down into four separate parts, of which the first three serves to further verify the robustness of previous analyzes (as shown in the work of H. Ozsoylev, J. Walden et al.) and the fourth is aimed at showing a connection between centrality and trading early. First, we investigated the effects of removing connections within brokerage houses (section 4.1). Our second task was to perform out-of-sample tests (section 4.2). We then continued by trying to calculate centrality measures when the connection window used when creating the connectivity matrix was extended from 30 minutes (which was the previous maximum) to 24 hours (section 4.3). Finally, in the main part of the project we investigated trader behavior during a number of large stock movements in relation to information events (section 4.4).

The rest of this chapter is structured as follows: each section is started off by explaining the task at hand in more detail, and this is then followed by analyzes (done by statistical regression methods) of the results from the program runs.

4.1 No brokerage house connections

At the ISE, all traders belong to brokerage houses. If there is a broker trading on behalf of his clients, there is a good chance that he will do the trades in a similar way. If the broker systematically prioritizes important clients, then these clients will be likely to appear to systematically trade in the same way even though they are not exchanging information (and should thus not be considered as connected according to our definition) - they just share the same broker. To account for this phenomenon, we did an analysis where connections between traders in the same brokerage houses were removed. This is easily implemented by comparing the brokerage house of traders before a connection is registered, and only register a connection if the corresponding brokerage houses are different.

Results

Table 1 shows the results of an Ordinary Least Squares (OLS) regression analysis of the measures computed in the program run. Collected measures include the number of trades N and trading volume V and profits for each trader, as well as the centrality measures – eigenvector centrality C and degree D – which were computed from the connectivity matrix.

The β-parameter of the regression denotes the correlation coefficient, while the t-parameter reveals the significance of the correlation; a value close to zero meaning no statistical significance.

As can be seen in table 1, the correlation between centrality and (standard as well as excess) profits is positive with a strong statistical significance. This is in line with previous analyzes analyses where connections in brokerage houses were allowed [9].

μ					μ^e				
	c	d	n	v		c	d	n	v
β_{OLS}	0.0107	-0.0131	0.0079	-0.0017	β^e_{OLS}	0.0103	-0.0133	0.0039	-0.0004
t_{OLS}	> 20	< -20	> 20	< -20	t^e_{OLS}	> 20	< -20	> 20	-6.5
$\Delta\mu_{OLS}$	2.2%	-2.6%	1.5%	-0.47%	$\Delta\mu^e_{OLS}$	2.1%	-2.6%	0.72%	-0.12%

Table 1: OLS regression analysis of parameters from a program run where the EIN was created with connection window Δt =30-minutes and no connections between traders in the same brokerage houses were allowed. Normalized profits (μ) and excess profits (μ^e) were regressed on log-centrality (c), log-degree (d) log-trades (n) and log-volume (v).

4.2 Splitting the data sample

Previous analyses extracted centrality measures and trader profits from the same data sample. One potential critique against this approach is that there may be an intrinsic correlation between measures drawn from the same data sample. To determine if the results are robust we perform out-of-sample tests where we split the available data set into two parts and then calculate the centrality measures from the first period and the normalized profits from the second. By doing this we avoid any endogeneity, and if the results are similar to the in-sample test, then one can conclude that the results are robust.

We did this using several configurations, where the split was placed after 4, 6 or 8 months. Thus, for the 8-4 split, centrality was computed from data concerning trades during the period January-August while profits were calculated from data involving the period September-December.

Results

The results of the different splits were quite similar. Table 2 shows the regression analyis on the results when the 8-4 split was used. For the analysis we had to restrict us to only using traders who traded in both parts of the year. In this case, a total of 228,538 traders were present in both samples.

As can be seen in table 2, centrality is positively correlated to (normalized) profits, with a coefficient 0.0110 and a strong statistical significance, as indicated

[9] H. Ozsoylev, J. Walden et al., 2012: "*Investor Networks in the Stock Market*", p.24

by the t-parameter being much larger than zero. The results are however not as strong in the case of excess profits, something which may be attributed to the short time period and smaller data sample.

μ					μ^e				
	c	d	n	v		c	d	n	v
β_{OLS}	0.0110	-0.0124	0.0057	-0.0009	β^e_{OLS}	0.0010	-0.0022	0.0029	-0.0006
t_{OLS}	10.7	-12.1	> 20	-7.3	t^e_{OLS}	1.2	-2.5	13.8	-5.3
$\Delta\mu_{OLS}$	1.8%	-2.0%	1.0%	-0.23%	$\Delta\mu^e_{OLS}$	0.2%	-0.4%	0.5%	-0.2%

Table 2: OLS regression analysis of parameters from an out-of-sample-test where centrality and degree were calculated during the first eight months, while profits, number of trades and trading volume were calculated during the final four months of the year. Connection window Δt =30-minutes was used. Normalized profits (μ) and excess profits (μ^e) were regressed on log-centrality (c), log-degree (d) log-trades (n) and log-volume (v).

4.3 Twenty-four-hour connection window

The length of the connection window being used when generating the connectivity matrix will obviously affect the number of registered connections in the connectivity matrix. Up until the start of this project, due to the memory and time limitations mentioned earlier, the longest connection window that had been used when constructing the connectivity matrix was 30 minutes. Since traders might share information with each other through phone calls, lunch meetings, email etc. a 30 minute connection window probably isn't enough to identify "real" connections. Information is assumed to diffuse rather slowly through the network, hence a better connection window would be somewhere around 24 hours. A window of that size would give two connected traders enough time to share a piece of information with each other, but at the same time not allow the information to become too public and thereby fully incorporated in the stock price.

As we know by now there are two main problems with a larger connection window; the matrix becomes too dense to fit in memory and the number of line comparisons in the deque grows exponentially, resulting in unreasonable calculation times. These problems have to be dealt with separately and we will start by addressing the second one.

Solving the time problem – preprocessing the data

The number of stocks being traded at the ISE is about 300, and one of the conditions for two traders to be considered as connected is that they have traded in the same stock. We will now assume that the trading in different stocks at the ISE is somewhat evenly distributed, i.e. on any given day the numbers of transactions involving each of the available stocks are about the same. This implies that, at any time, the fraction of lines in the deque that deals with a certain stock would be about $1/300$. Thus, when comparing a line in the deque with all others, around 299/300 of the comparisons will not result in a connection. Clearly this is a lot of unnecessary work being done since comparing two lines in the transaction data is a relatively costly operation. The question becomes; how to get rid of the unnecessary line comparisons? The answer lies in preprocessing the data.

By dividing the raw data into separate files before searching for connections, with each file only containing trades in a specific stock, the number of lines, and thereby comparisons, in the deque are dramatically reduced. The downside of this is that the raw data has to be split, which in our case took approximatively one week on a single CPU. This only needs to be done once however, after which the program can be run much more efficiently, allowing for the possbility to use much longer connection windows than before.

A side-effect of this approach is that we end up with a large number of separate connectivity matrices, each containing connections generated from trades in a specific stock. Of course, in the end we still want to have a complete connectivity matrix generated from trades in all stocks, hence we need to merge these matrices. This is where we encounter a memory problem since the total number of connections being identified still will be the same; the difference being that the connections are now spread amongst several matrices.

Solving the memory problem – merging the matrices

A way to overcome the memory problem is to divide the total matrix into k×k sub-matrices and construct it one sub-matrix at a time. By summing the corresponding elements of all stock-matrices, for each sub-matrix, the total matrix is built piece by piece. Before moving on to the next sub-part of the total matrix, the threshold M is applied to the current sub-part in order to eliminate all connections below a certain value. In this way we get rid of noise connections and at the same time reduce the density of the matrix. Without the threshold the total matrix would soon become too dense to fit in memory. Note that the purpose of using a larger connection window is not necessarily to capture a larger number of connections but to single out connections, by using the threshold, between traders that have consistently traded in the same way. When this filtering has been done, the current sub-part of the total matrix is finished and the corresponding elements of the next sub-part can be summed. Finally, when all sub-matrices have been created, filtered and added, what's left will be the total matrix containing only the persisting connections, and the program can continue as before by calculating centrality measures etc.

Unforeseen problems

When dealing with such an extensive dataset as we are here, implementation is far from straightforward. Even simple operations such as the summing of two matrices will yield problems if one is not careful when implementing them. Therefore, all algorithmic operations need to be refined in a way that will work both from a time and memory perspective. This was easily the most time consuming part of the project and a lot of tedious work were put into making it work. The algorithm and the modifications that were introduced to deal with the memory and time limitations worked perfectly well on test data. However, when applied to the whole data set with a 24-hour connection window, no results could be obtained. We do not know the exact reason for this but believe it has something to do with memory handling when the matrix becomes too dense. We tried numerous approaches in which we tweaked the algorithm in different ways without being able to produce any results.

We did however manage to produce sub-matrices in some of our program runs. One of these, constructed from the first three months of the dataset, was then used as a connectivity matrix containing a third of the traders, which we were then able to use in further calculations. Even this "small subset" of the full data resulted in a matrix size of approximately 10 Gb, and all the computational work in creating the matrix and calculating centrality measures etc. took more than a day on the Kalkyl system's processors (not counting the time to preprocess the data).

Results

As previously mentioned, we were unable to obtain a connectivity matrix based on data from the full year. Using time-truncated data, we were however able to obtain a sub-matrix which was then used in further calculations. Table 3 shows a multivariate regression on the results of the program run where the EIN was created from the first three months of the transaction data, including only the first third of the traders (approximately 193,000). Similarly, table 4 shows results from a univariate regression.

Both tables show that there is a positive correlation between centrality and normalized profits (standard as well as excess), with a strong statistical significance (t-statistics being larger than 20 in three cases, and 15.7 in one).

μ					μ^e				
	c	d	n	v		c	d	n	v
β_{OLS}	0.065	-0.063	0.0078	0.00002	β^e_{OLS}	0.020	-0.020	0.003	0.000006
t_{OLS}	> 20	< -20	> 20	-1.5	t^e_{OLS}	15.7	-15.5	17.8	0.55
$\Delta\mu_{OLS}$	7.7%	-7.6%	1.5%	-0.04%	$\Delta\mu^e_{OLS}$	2.4%	-2.4%	0.5%	0.01%

Table 3: Multivariate OLS regression. Normalized profits, μ, and excess profits, μ^e are regressed on log-centrality (c), log-degree (d) log-trades (n) and log-volume (v). The EIN was created using a time window of $\Delta t = 24$ hours. The sample is restricted to one third of the investors (approx. 193,000), and centrality and degree is calculated using the first three months of trades.

μ						μ^e					
	c	d	$c-d$	n	v		c	d	$c-d$	n	v
	0.0091						0.0028				
		0.0083						0.0025			
			0.0028						0.0009		
				0.0071						0.0023	
					0.0054						0.0046
t	> 20	> 20	> 20	> 20	> 20	t^e	> 20	> 20	7.7	> 20	> 20
$\Delta\mu$	1.1%	1.0%	0.3%	1.4%	1.0%	$\Delta\mu^e$	0.3%	0.3%	0.01%	0.5%	0.4%

Table 4: Univariate regression. Normalized profits, μ, and excess profits, μ^e are regressed on log-centrality (c), log-degree (d) log-trades (n) and log-volume (v). The EIN was created using a time window of $\Delta t = 24$ hours. The sample is restricted to one third of the investors (approx. 193,000), and centrality and degree is calculated using the first three months of trades.

4.4 Information Events

When a news story concerning a listed company is published it will most likely affect the company's stock price. In fact, more often than not, a sudden movement in a company's stock is the result of a news publication. This should be no surprise, but what if you got hold of a news story before the public did? It would surely give you a great opportunity to make a profit, since you in a way would be able to foresee the future development of the stock. Furthermore, the publication of a news story is often preceded by rumors and we assume that centrally located traders in the information network, due to their vast number of connections, should receive these rumors earlier than non-central traders and be able to benefit from it. In this study, our tests are purely based on trades, which in itself is not sufficient to prove that trading early in relation to a news event is a result of information diffusion. However, if we can show that there is a connection between trading early and being central in the EIN, this will support the theory that central traders receive valuable information earlier than non-central traders. To examine this, we investigate occasions where news about a listed company has been published, which can be linked to a sudden movement in that company's stock. These sort of situations are ideal since the information regarding the company eventually becomes public, but rumors and such might have circulated during the days leading up to it. We were given around 150 news events that could all be linked to sudden stock movements. The events consisted of both positive and negative news, two examples are:

- Positive event – Besiktas
  In July 2005, the football club Besiktas made a public statement saying they considered acquiring a new player. After the news was published, the club's stock rose since this was regarded as positive for the club.

- Negative event – SekerBank
  In July 2005, the bank SekerBank made a public statement announcing that they were about to sell a majority of its shares at a price significantly lower than the market value. This clearly was a negative news story and resulted in a sudden fall of the stock price.

In both these cases, it is possible that rumors circulated before the statements were announced. An indication that traders have received rumors regarding a company is that they have traded in the company's stock prior to the stock move. If we could show a correlation between trading early and being central in the network, then this would support the main hypothesis of the project – that information is diffusing through the network and that well-connected traders receive information early and benefit from it.

So, to investigate the behavior of traders, an event window is specified as a number of days before and after a stock move. The idea is to investigate all transactions within the event window, for each event, and register how early traders act compared to the stock move. Data will only be registered for traders that have traded in the right direction, i.e. either bought or sold. If a news story is considered to be positive, an informed trader will probably buy stocks in the affected company since he expects the stock to rise, and in the same way he will obviously sell if the story is a negative one.

Although the most interesting thing to measure is how early traders act, it is not sufficient to only register the date of the first trade within the event window, since this will rank uninformed traders as informed if they just happen to trade early. Specifically, uninformed traders who are trading often and in a lot of different stocks might be classified as very central since they probably, at some time, will trade early in the stocks concerned by the news events. To deal with this we will record all transactions a trader performs in the current stock within the event window, and calculate a mean trade day. This will not bias our results since uninformed traders that trade a lot are equally likely to also trade after the stock move, resulting in a mean trade day close to the stock move. A problem with this approach is that some informed traders might, for a number of reasons such as different trading strategies or the current state of the market, spread their transactions among several days instead of executing them all at once. However, if they are indeed informed they will spread their trades among the early days of the event window, still giving them an early mean trade day. Aside from this, information about the last trade made in the window, the number of transactions and the aggregate value of all trades are extracted.

As a side-note we also verified that trading early leads to profits by calculating the profits of all transactions in the event windows that takes the same side as the event itself (i.e. buying prior to a positive event and selling prior to a negative event). However, this should be of little interest since the trades by definition should lead to profits, since we only record trades in the "profitable" direction (i.e. sells before a price fall and buys before a rise).

When studying the raw data, one can see that the market sometimes will react before the corresponding news story is published. In those cases a possible explanation might be that the information has already become somewhat public, through rumors or alternative media, prior to the news being published. However, this should not affect our predictions since central traders would still receive the information earlier than those that are not.

We identified eleven information events that concerned appropriately sized companies (not too small, as to to avoid low stock liquidity which leads to problems in timing and pricing when there aren't many buyers of the stock, and not too large either, since the rumours may then spread too quickly), which we based our investigations on. A short description of these events are found in table 7 in the appendix. We also did robustness tests where we included more events, spread out through the year, which lead to a total of 24 events, as described in table 8.

Results

The parameters from the regression of the results from our program run on the previously described eleven events are shown in table 5. The table shows that the average trading time is negatively related to (the logarithm of) rescaled centrality with a strong statistical significance (t much smaller than zero), which means that central traders trade earlier in regard to the information events.

$c-d$			
	T	n	v
β_{OLS}	-0.000007	-0.00006	-0.00001
t_{OLS}	< -20	< -20	-14

Table 5: Log-rescaled centrality, $c - d$, regressed on average trading time, t, log-number of trades, n, and log-volume, v. The trade data was computed only during the event windows among the eleven events defined in table 7.

In order to verify the robustness of these results, an extended event list as defined in table 8 was used. The resulting regression of these results is found in table 6, which shows that there is a statistically significant relation between centrality and trading early also in this case.

c		$c-d$		
	T	T	n	v
β_{OLS}	-0.021	-0.000002	-0.00007	-0.000008
t_{OLS}	-16.6	-13.0	< -20	-12.4
$t_{t-error}$	-7.9	-5.5	< -20	-6.7
t_{Ramsey}	-6.3	-13.0	< -20	-12.4

Table 6: Centrality regressed on trading time for extended set of information events (24 events). Average trading time is negatively related to centrality, both in univariate (c on T) and in multivariate ($c - d$ on T, n and v) regressions.

5 Conclusions

This paper, which has a specific focus on information events, is part of a larger project that extends previous work published on information diffusion in stock markets. By investigating raw transaction data from the Istanbul Stock Exchange during the year 2005, an Empirical Information Network (EIN) is constructed and acts as a proxy for the real information network. All stock transactions during the year 2005 is a huge data set, and working with it requires a careful implementation – all algorithmic calculations need to be refined in such a way that they meet time and memory limitations. In our (and previous) implementations, profits are calculated from trades throughout the year, for each trader, in a way that doesn't take into account a trader's portfolio entering the year, and these profits have been found to be positively correlated with centrality calculated using the EIN. We have through this project been able to further verify the robustness of these results by showing similar, statistically significant positive correlations between centrality and profits in the cases where:

- connections between traders within the same brokerage house are discarded in order to remove any influence from stock brokers trading on behalf of their clients
- centrality and profits are calculated from separate data samples (out-of-sample tests) in order avoid any endogeneity in the measures
- the connection window is extended from 30 minutes to 24 hours, in order to accomodate realistic information diffusion

In the main part of the project, we investigated whether centrality in the EIN was correlated to trading early on information concerning news events. We found this to be the case, with strong statistical significance, which goes some way to support the theory that traders with a lot of connections, i.e. the central agents, receive and act upon valuable information before uninformed traders do, which as a consequence leads to higher profits. We also examined the robustness of these results by extending the set of events (from 11 to 24), and the results were similar and statistically significant.

We thus conclude that there is strong empirical support for the theory that information diffusion plays an important part in stock market trading, and that being well-connected gives you a better chance of making good investment decisions.

6 Acknowledgements

First, we would like to thank our supervisor – prof. Johan Walden – for giving us the opportunity to work this very interesting project, for giving excellent feedback throughout the course of the project, and for providing us with all the tables found in this report and the regression analyzes therein. We would also like to thank Niclas Eriksson and Ludvig Larruy for providing us with their code and for helping us during the start-up of the project. Finally, we would like to thank Maya Neytcheva for the support, and for helping us with our UNIX and UPPMAX questions.
Thanks!

7 References

H. Ozsoylev, J. Walden et al., 2012: "*Investor Networks in the Stock Market*" (working paper)

M. E. J. Newman: "*The mathematics of networks*"
http://www-personal.umich.edu/ mejn/papers/palgrave.pdf
(Retrieved 2011-12-04)

N. Eriksson, L. Larruy, 2010: "*Network analysis of a stock market*"

R. J. Shiller, J. Pound, 1989: "*Survey evidence on the diffusion of interest and information among investors*", Journal of Economic Behavior, vol.12

8 Appendix: Event details

Ticker	*Name*	*Main operation area*	*Movement date*	*Sign and magnitude*
BJKAS	Besiktas Futbol Yatirimlari	Soccer	7/26/2005	+19.4%
			12/26/2005	-33.5%
DEVA	Deva Holding	Pharmaceuticals	7/12/2005	+15.8%
DYOBY	DYO Boya	Paint and chemicals	7/14/2005	+12.25%
EREGL	Eregli Demir ve Celik Fabrikalari	Steel manufacturing	9/12/2005	+10.56%
SAHOL	Haci Omer Sabanci Holding	Multibusiness enterprise	10/4/2005	+6.99%
SEKFK	Seker Finansal Kiralama	Financial leasing	7/8/2005	-21.27%
SISE	Turkiye Sise ve Cam Fabrikalari	Glass manufacturing	7/29/2005	+7.08%
SKBNK	Sekerbank	Banking	7/12/2005	+14.55%
TEKST	Tekstilbank	Banking	8/16/2005	+12.27%
TNSAS	Tansas	Retail	8/19/2005	+12.12%

Table 7: Stocks and daily price movements used in the analysis of the relationship between centrality and information events.

Ticker	*Movement date*	*Ticker*	*Movement date*	*Ticker*	*Movement date*
ASELS	3/9/2005	DOHOL	1/14/2005	NTHOL	4/7/2005
GIMA	4/22/2005	KARTIN	1/25/2005	TEBNK	1/12/2005
KRDMA	5/17/2005	TSKB	1/14/2005	TUPRS	3/18/2005
ULKER	2/14/2005	AKGRT	2/5/2005	VESTL	2/14/2005
BFREN	2/5/2005	BFREN	1/25/2005	DOYBY	7/14/2005
BOYNDR	1/18/2005	ECILC	3/14/2005	TEKST	8/16/2005
TNSAS	8/19/2005	DEVA	7/12/2005	EREGL	9/12/2005
SAHOL	4/10/2005	SISE	7/29/2005	SKBNK	7/12/2005

Table 8: Expanded set of information events, reported in public news outlets within 7 business days before and after the stock movement dates.