Complex Networks Analysis with Transfer Entropy

İbrahim Çağlar

Doctor of Philosophy
University of York
Computer Science
2021

I dedicate my thesis work to my esteemed parents *Hatice* and *İsmail*, who have never spared their unconditional love and support, and to my brother *Mustafa* and his children *Hatice Kübra* and *İsmail Hamza*.

I also dedicate this thesis to my wife, *Dr. Durdane Çağlar* , who has been a constant source of support and encouragement during the challenges of my PhD an life.

Abstract

Nowadays, data analysis has become more complicated.Agencies such as social media and online news sites cause rapid dissemination of information. This situation creates a lot of relevant and irrelevant data. It takes a lot of effort to make a meaningful analysis by extracting as many causal relational data as possible from the others. For example, from the diseases that trigger each other or the effects of the bankrupt company on other market players. In our thesis, we made a study to make these complex networks more understandable and readable.

We tried to apply Schreiber's transfer entropy on a complex network as a way to characterise interaction between data, in other words, information flow. We measures network similarity using Jensen-Shannon divergence and Kullback-Leibler divergence. With this, we wanted to compare the distribution of correlations between different networks. We explore how both weighted and unweighted representations derived from these characterisations perform on real-world time series data. We also use the transfer entropy to weight the edges of a graph where the nodes represent time series data and the edges represent the degree of commonality of pairs of time series. We also make a comparison between the graph characterisation calculated by von Neumann entropy and transfer entropy.

We also worked on smoothing the edge entropy by applying diffusion operation on how this information flow can be in multiplex graphs. We examined the causality of this information flow, especially in time-varying multiplex graph.

Acknowledgements

I would like to express my sincere gratitude to everyone who has supported me throughout this journey.

I would like to thank Professor Edwin R. Hancock for being my supervisor for his patience, advice, and support in helping me complete this PhD. I would be glad to thank Professor Richard C. Wilson for his support and advice as we conclude this journey. I would also like to thank committee members Vincenzo Nicosia and Adrian Bors for their support and advice. I would also like to thank my family and friends for their constant love and support. In addition, I would also like to thank my friends, colleagues, department faculty and staff for making my time at York University a wonderful experience.

Finally, I would like to thank the Republic of Turkey Ministry of National Education for the support they have given me to do this PhD.

Declaration

I declare that the work in this thesis is only my own, except in cases where it is attributed and referred to another author. Most of the materials in this thesis have been previously published by the author. A list of publications can be seen later in the page.

İbrahim Çağlar

List of Publications

- Ibrahim Caglar, Edwin R. Hancock: Graph Time Series Analysis Using Transfer Entropy. S+SSPR 2018: 217-226
- Ibrahim Caglar, Edwin R. Hancock: Network Time Series Analysis Using Transfer Entropy. GbRPR 2019: 194-203

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we provide for introduction to the thesis. In particular, we give our motivation and goals while doing this research. The chapter commences by introducing some ideas and problems for graph characterisation in the network science literature. Afterwards, we explain the characterisation method we recommend to overcome these difficulties. The chapter finishes by giving an outline of the reminder of the thesis.

1.1 Problem

For many years, people have tried to establish a cause-effect relationship between their observations, experiences and existing conditions. They observed eclipses, stars, earthquakes to better understand the factors affecting their lives. People saw these factors as a reason for good or bad events. In the simplest sense, they wanted to change these results by praying, offering sacrifices and various rituals for the next year's crop or the victory of the next war. Later, people focused more on understanding how the World works and on the connection between effect and factor, as in studies such as Pavlov's Dogs or Pasteur's experiments. In modern times, this search continues to expand with more systematic foundations such as forecasting the weather, the causes of diseases or financial market analysis etc.

Network analysis is the analysis of interactions between entities, which can

be products, customers, diseases, evolution or even devices. Organisations and companies from all over the world use network analysis to demonstrate an effective approach in marketing, optimisation, fraud detection, disease detection and investment analysis [4, 15, 34]. The Swiss mathematician Euler, who solved the Königsberg Bridge Problem in 1735, opened a new door for science by making the first systematic study of network analysis [63].

Over time, network analysis has evolved from a puzzle to problems involving petabytes of data analysis for instance weather forecast, and customer-product simple analysis. When the ease of data collected by digital materials such as mobile phones and computers and the size of this data are taken into consideration, the already complex system has become a more complex structure. Not only the complexity of the data, but also its size is increasing day-by-day. This has made network analysis more complicated, complex and costly than ever. In the last two decades, the acceleration of the internet and its widespread use have affected social structure and habits. Access to information became easier and access to products became global rather than local. Wherever in the world, the entertainment industry has transformed into video-on-demand services, games and social media sharing. Spending turned online and investments turned into automated trading/scripts. All of these new habits leave a variety of data behind them that can be searched. Scientists are trying to analyse and make sense of these complex systems and big data as quickly as possible. Because timely analysis may save lives, may help farmers or help to make a more profitable trade.

Here we will focus more on the analysis of financial data. Significant amounts of data are generated from auto-trade [67], online applications and growing buy-sell orders [61]. It has become very easy for people to make trade decisions with the information they have obtained from both conventional media and new types of social media communication systems, and to put these decisions into action immediately [50, 70]. It is becoming difficult for analysts and scientists to find a pattern among this complex data.

1.2 Goals

The aim of this study is to analyse complex structures in financial networks and to try to characterise them with methods based on statistical principles. We will model financial data to with the companies in the stock market as nodes and consider the relations between these companies as edges.

We will create these edges both with cross-correlation, which is a widely used in field, and with transfer entropy. The main reason for using transfer entropy is that it maintains causality while establishing the relationships between nodes. Causality is important in finance because two companies may not affect each other equally, and we would like to include this feature in our model.

What we are trying to do to examine the similarities of the movements of stocks in times of crisis and in times of non-crisis. We will measure whether other stocks are affected by the movements of a stock, and if it is, how much it is affected. At each time epoch we construct a weighted graph in which the edge weights are computed from transfer entropies between pairs of nodes. This is an instantaneous snap shot of the pattern of information flow between nodes. We analyse time series by observing how this network structure evolves with time. We have used statistical methods such as Principal component analysis (PCA), Multidimensional scaling (MDS), Jensen–Shannon divergence (JSD) to show this evolution more clearly. We have attempted to reveal the information flow between the nodes by applying directional heat diffusion on Multiplex Network representing of stock market data. We performed to make an analysis by including information from the past and present in the model.

1.3 Thesis Overview

This Thesis is organised as follows. In Chapter 2, we review the research literature related to the study presented in the thesis and give the necessary brief descriptions of the concepts invested . In Chapter 3, we construct a network model using transfer entropy. We also compare the model constructed using approximate von Neumann entropy with the model constructed using Transfer Entropy. We

also give the necessary background about graph theory in this section. Chapter 4 express to use of heat kernel smoothing on single and multiplex graphs. We build a structure by based on the information flow as a heat flow. We perform an analysis on both a single layer graphs and a multiplex graphs. Finally, in Chapter 5, we discuss a conclusions from the work presented We identify the novel contributions made as a result of the study. We also discuss several limitations and possible future research directions.

Chapter 2

Literature Review

In this chapter we will review the characterisation the general problem of characterisation of data using graph-bases representation. We will also give the basic definitions that we use based on graph theory. In addition, we will explain how concept transfer entropy has developed over time and the growth in its application areas. We will also give the definition of von Neumann entropy review the literature. We also focus on heat diffusion on graphs and its basis in information theory.

2.1 Graph Theory

Graph Theory has its origins in 1736, when Leonhard Euler solved the Königsberg Bridge problem. His work is commonly quoted as the origin of graph theory [63]. Almost a century and a half later, the term "graph" was introduced by J.J. Sylvester in 1878 [75]. In general terms, it is a mathematical structure formed by nodes and edges. It creates the mutual relations of a pair of objects. It can be used to represent and solve problems from airline network optimisation to neural network calculations.

Definition 1 (Graphs)**.** *A graph G is denoted as $G = (V, E)$, where V is the finite set of vertices and $E \subseteq V \times V$ is the finite set edges in the graph.*

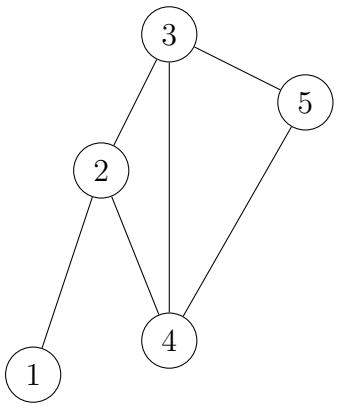

Figure 2.1: An example of a graph with 5 nodes and 6 edges.

The vertices correspond to the dots in figure 2.1, and the edges correspond to the lines between them. For Figure 2.1; $G = (V, E)$ where, $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{3, 5\}, \{3, 4\}, \{4, 5\}\}$. Note that since the sets are undirected, $\{\{1, 2\}$ and $\{\{2, 1\}$ are different definitions of the same edge. Simple graphs do not contain directed edges such $\{1, 2\}$ and $\{2, 1\}$ can be used interchangeably.

Definition 2 (Directed Graphs)**.** *A directed graph or digraph is a set of vertices and a collection of directed edges that each connects an ordered pair of vertices.*

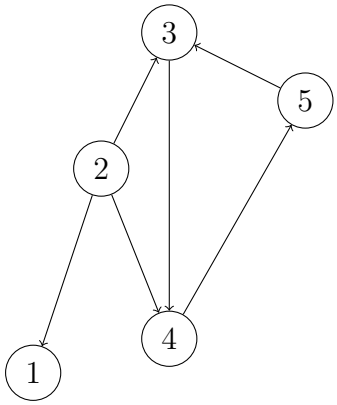

Figure 2.2: An example of a digraph with 5 nodes and 6 directed-edges.

The vertices correspond to the dots in figure 2.2, and the edges correspond to the lines. The arrows on the line indicates the direction. For Figure 2.2; $G = (V, E)$ where, $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{2, 1\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}, \{5, 3\}\}$. Note that since the sets are directed, $\{1, 2\}$ and $\{2, 1\}$ have different meanings and they cannot be used interchangeably.

Definition 3 (Weighted Graphs)**.** *A weighted graph is a graph with weights assigned to each edge. A weighted graph consists of a graph* $G = (V, E)$ *and a weight function* $\omega : E \to \mathbb{R}$*. Usually, the edge weights are non-negative real numbers.*

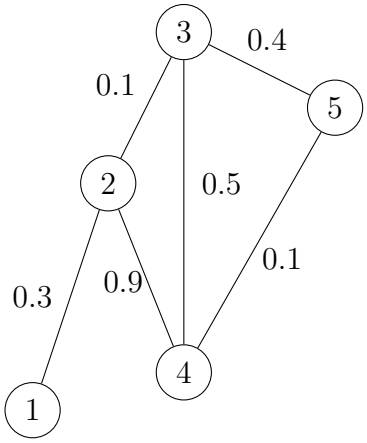

Figure 2.3: An example of a weighted graph with 5 nodes, 6 edges and 6 weight corresponds to 6 edges.

For example, we might be interested in the distance of a highway between a pair of cities, the correlation between a pair of stocks. These values can be by the weight between two nodes. Weighted graphs may be directed or undirected. If an egde is directional, there can be one weight, and if it is bi-directional, there can be two different weights with different values. As an example, a vehicle travelling from a high altitude to a low altitude will need less fuel then while trying to climb. If we take the two cities as nodes and take the roads as edges, the fuel usage is considered as a weight.

Definition 4 (Line Graph)**.** *The Line Graph represents the adjacencies between edges of* G*. A line graph obtained by two vertices of* $LG(G) = (V_L, E_L)$ *are adjacent if and only if their corresponding edges of* G *have a vertex in common, where;*

$$V_L = E$$
$$E_L = \{(u, v), (v, w) : (u, v) \in E, (v, w) \in E\}$$

Definition 5 (Adjacency Matrix)**.** *Given a graph* $G = (V, E)$ *where,* $V =$

$\{v_1, v_2, ..., v_n\}$, *the adjacency matrix for* G *is the* $n \times n$ *matrix* $A = \{a_{ij}\}$ *where,*

$$a_{ij} = \begin{cases} 1 & if\{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

If G *is a weighted graph with edge weights given by* $\omega : E \to \mathbb{R}$*, then the adjacency matrix for* G *is* $A = \{a_{ij}\}$ *where,*

$$a_{ij} = \begin{cases} \omega(\{v_i, v_j\}) & if\{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0.3 & 0 & 0 & 0 \\ 0.3 & 0 & 0.1 & 0.9 & 0 \\ 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0.9 & 0.5 & 0 & 0.1 \\ 0 & 0 & 0.4 & 0.1 & 0 \end{bmatrix}$$

Adjacency Matrix for the unweighted graph in Figure 2.1.

Weighted Adjacency Matrix for the weighted graph in Figure 2.3.

Figure 2.4: Adjacency Matrix representation for the given two graphs.

Definition 6 (Degree Matrix)**.** *Given a graph* $G = (V, E)$ *where,* $V = \{v_1, v_2, ..., v_n\}$*, the degree matrix for* G *is the* $n \times n$ *diagonal matrix defined as,*

$$D_{i,j} = \begin{cases} deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where $deg(v_i) = |\{v_j \in V | (v_i, v_j) \in E\}|$*, and counts the number of times an edge terminates at that node.*

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Figure 2.5: Degree Matrix representation for the graph in Figure 2.1.

Definition 7 (Incidence Matrix)**.** *Incidence Matrix is the matrix that shows the relationship between rows and columns in a graph. Given a graph* $G = (V, E)$ *where, number of vertices* $|V| = n$*, and number of edges* $|E| = m$*. The incidence matrix is* $n \times m$ *matrix, whose entries are as follows*

$$IM_u(v, e) = \begin{cases} 1 & \text{if } v \text{ is the incident node of edge } e \\ 0 & \text{otherwise} \end{cases}$$

$$IM_d(v, e) = \begin{cases} 1 & \text{if } v \text{ is the terminal node of edge } e \\ -1 & \text{if } v \text{ is the initial node of edge } e \\ 0 & \text{if } v \text{ is not in } e \end{cases}$$

$$IM_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \qquad IM_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Incidence matrix for the undirected graph in Figure 2.1.

Incidence matrix for the directed graph in Figure 2.2.

Figure 2.6: Incidence matrix for undirected and directed graph.

Since large-sized graphs are difficult to draw and read, we can determine out

which nodes are connected to which edge using the incidence matrix. If our graph is a directed graph, we can also determine the direction of an edge for the incidence.

Definition 8 (Laplacian Matrix)**.** *Given a graph $G = (V, E)$ where, $V = \{v_1, v_2, ..., v_n\}$, the Laplacian matrix L is defined as $L = D - A$ where D is the degree matrix and A is the adjacency matrix of the graph. The element-wise definition of the Laplacian matrix is;*

$$L_{i,j} = \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 2.7: Laplacian Matrix for the Figure 2.1 $L = D - A$.

The Laplacian matrix captures with many useful properties of a graph. The spectral decomposition of the Laplacian matrix allows the creation of low-dimensional embeddings of the nodes into a vector-space [1, 17].

Definition 9 (Normalised Laplacian Matrix)**.** *Given a graph $G = (V, E)$, the normalised Laplacian matrix $\widetilde{L}$ is defined as $\widetilde{L} = D^{-1/2}LD^{-1/2}$ where L is the Laplacian matrix and D is the degree matrix of the graph. The element-wise definition of the Laplacian matrix is;*

$$\widetilde{L} = \begin{cases} 1 & \text{if } v_i = v_j \text{ and } deg(v_i) \neq 0 \\ \frac{-1}{\sqrt{deg(v_i)deg(v_j)}} & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Note that the normalised Laplacian $\widetilde{L}$ is a symmetric matrix because L and $D^{1/2}$ are symmetric. The spectral decomposition of the normalised Laplacian matrix is $\widetilde{L} = \sum_{i=1}^{|V|} \lambda_i \phi_i \phi_i^T$ where λ_i are the eigenvalues and ϕ_i the corresponding eigenvectors of $\widetilde{L}$.

A node with a large degree will have a greater impact on the matrix properties in the Laplacian matrix than the remaining nodes. Normalisation aims to make remove the bias towards large degree nodes.

Definition 10 (Principal component analysis (PCA))**.** *Principal component analysis is a method of size reduction while preserving as much information as possible. Consider an $n \times n$ data matrix X. Size reduction is achieved with the following steps;*

Step 1 - Normalise the data

$$z = \frac{x - \mu}{\sigma}$$

μ is the mean of the column from each entry. σ is the standard deviation of that column.

Step 2 - Calculate the Covariance Matrix for the normalise the data

$$\textit{Covariance Matrix (CM)} = \begin{bmatrix} Cov(x_1, x_1) & \dots & Cov(x_1, x_n) \\ \dots & \dots & \dots \\ Cov(x_n, x_1) & \dots & Cov(x_n, x_n) \end{bmatrix}$$

Step 3 - The eigenvectors and eigenvalues of the covariance matrix
The eigenvalues of CM are roots of the characteristic equation

$$\det(CM - \lambda I) = 0$$

$$\det\left(\begin{bmatrix} Cov(x_1, x_1) & \dots & Cov(x_1, x_n) \\ \dots & \dots & \dots \\ Cov(x_n, x_1) & \dots & Cov(x_n, x_n) \end{bmatrix} - \begin{bmatrix} \lambda & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & \lambda \end{bmatrix} \right) = 0$$

$$\det\left(\begin{bmatrix} Cov(x_1,x_1)-\lambda & \dots & Cov(x_1,x_n) \\ \dots & \dots & \dots \\ Cov(x_n,x_1) & \dots & Cov(x_n,x_n)-\lambda \end{bmatrix}\right) = 0$$

After solving this equation for the value of λ. The eigenvectors (ϕ_i) corresponding to the eigenvalues (λ_i) can be calculated.

Step 4 - Feature Vector

The feature vector is that contains the eigenvectors corresponding to the eigenvalues in descending order as columns.

$$Feature\ Vector = \begin{bmatrix} \phi_1 & \dots & \phi_k \end{bmatrix}$$

where k represents the number of components needed.

Step 5 - Forming Principal Components

$$Reduced\ Data\ Set = Feature\ Vector^t \times Normalise\ Data\ Set^t$$

PCA is a method for reducing the dimensionality of large data sets but at the same time minimizing information loss. Principal components are orthogonal because they are eigenvectors of a covariance matrix [43].

2.2 Entropy

Entropy is a physical property that indicates the degree of disorder or uncertainty in a system. Entropy is such a broad concept that we can see it from the diagnosis of asthma [81] , as well as in the energy released after The Big Bang [16]. We can also encounter the concept of entropy in information theory and algorithms [72]. Entropy first entered the literature as a thermodynamics concept [79]. However, in the 1950s, due to the development and expansion of computers, it began to be examined more as a statistical value. In this thesis, we will deal with Entropy within the scope of information theory.

In the late 1940s, Claude E. Shannon write the seminal paper called "A Mathematical Theory of Communication"[72]. Shannon focused in how best to encode

the information a sender to transmits. He has determine the minimum number of the bits needed to encode information in a signal. Here, entropy measures information lost from the system. It is the measure of data loss before the receiver receives a data. After this study, the concept of entropy used in information theory is often called Shannon entropy.

Definition 11 (Entropy)**.** *The entropy is the average level of "information" contained or "uncertainty" specific to a variable's possible outcomes. The entropy for an event X is defined as;*

$$\begin{aligned} H(X) &= \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)} \\ &= -\sum_{x \in X} p(x) \log_2 p(x) \end{aligned} \tag{2.1}$$

where, resummation is over the variable's possible values. X random variables with probability distributions $p(x)$, $x \in X$.

The base of the logarithm may vary to suit the desired applications. If the base is 2, gives the units are *bits*[49], while with base e the units are *nats* and base 10 the units are *dits*[33]. Entropy is a function of the distribution of an event. It does not depend on the actual values, but only on their probabilities.

An entropy can also be calculated for two discrete random variables. The entropy that can be calculated for two variables without any conditions is called joint entropy.

Definition 12 (Joint Entropy)**.** *The joint entropy of X and Y is defined as*

$$\begin{aligned} H(X,Y) &= \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{1}{p(x,y)} \\ &= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(x,y) \end{aligned} \tag{2.2}$$

where, $p(x,y)$ is the joint probability of these values occurring together. The sets X and Y contain the two discrete random variables with probability distributions $p(x)$, $x \in X$ and $p(y)$, $y \in Y$.

The joint entropy of a set of random variables is a non-negative number ($H(X,Y) \geq 0$). Due to properties of joint probability $H(X,Y) = H(Y,X)$. The joint entropy of a set of variables is greater than or equal to the maximum of the individual entropies of all the variables in the set i.e. $H(X,Y) \geq max[H(X), H(Y)]$. Similarly, the joint entropy of a set of variables is less than or equal to the sum of the individual entropies of all the variables in the se i.e. $H(X,Y) \leq H(X)+H(Y)$. Equality has to if and only if X and Y are statistically independent.

If an observation or condition comes into play when calculating the entropy of two random variables, the entropy is called conditional entropy and is defined as follows.

Definition 13 (Conditional entropy)**.** *The entropy of Y when X is known [23]*

$$\begin{aligned} H(Y \mid X) &= \sum_{x \in X} p(x) H(Y \mid X = x) \\ &= -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2 p(y \mid x) \\ &= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(y \mid x) \\ &= \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x)}{p(x,y)} \end{aligned} \tag{2.3}$$

In other words, it is the expected number of bits required to identify X when Y is known to both encoder and decoder.

The conditional entropy equals zero if and only if the value of Y is completely determined by the value of X. On the contrary, iff Y and X are independent random variables, $H(Y \mid X) = 0$. Also, according to the chain rule [23] the joint entropy of a pair of random variables is the sum of the entropy of one and the conditional entropy of the other ($H(X,Y) = H(X) + H(Y \mid X)$).

The Relative Entropy or Kullback–Leibler Divergence (KLD) is a measure of the distance between two distributions or a measure of how much one probability distribution differs from a second reference of probability distribution.

Definition 14 (Relative Entropy or KLD). *P and Q are two probability distributions defined on the same probability space. The Kullback–Leibler divergence between P and Q is defined as [23]*

$$D_{KL}(P \parallel Q) = \sum_{x} P(x) \log_2 \frac{P(x)}{Q(x)} \tag{2.4}$$

From their definition it follows that the KLD is not a symmetrical distance.

$$D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$$

Moreover $D_{KL}(P \parallel Q) \geq 0$, and also iff $D_{KL}(P \parallel Q) = 0$ when $P(x) = Q(x)$. As a result, KLD does not follow to triangle equality, and so it is not a true distance.

Entropy has been used in various ways in a variety studies. Demetrius and Manke used entropy analysis in biology to formulate an evolutionary model using correlation measures [29]. Their model based on the Darwinian principles of evolution. They represented molecular entities such as proteins as nodes, and the interactions between these entities as edges. Their result showed the entropy of network is correlated with robustness the entropy reaches maximal values where evolution increases robustness, and reaches minimum values where the evolution decreases in robustness.

Cancer networks can also be representable using graphs and entropy [68]. According to Tannenbaum et al., search engine and cell biology have common elements such as noisy data and reliance on input [76]. They have shown that search engines and cell biological cycles are based on empirical distributions. Their approach has been used to assess financial market robustness and to differentiate the biological networks of cancer cells from healthy ones [32].

Regardless of the field of study, if a system can be represented as a graph, the entropic approach to this system gains meaning. Low or high value of entropy can interpreted of the structure of the network representation.

2.3 Information Transfer for Finance

Research of time series analysis is a mature discipline spanning many decades. It can occur in different forms, from analysis of the movements of objects in space [11] to stock market analysis[13, 34]. Lacasa et al. introduced a non-parametric method to analyse multivariate time series [51]. This method enhances the information for high dimensional dynamic systems it can be used for the analysis of large, non-stationary and heterogeneous time series. The authors also study financial time series to demonstrate that can account for their method the US's 35 largest company stocks in NYSE and NASDAQ over the period 1998 - 2013.

In stock market analysis, transfer entropy is frequently used to measure causal relationship. It is easy to determine which stock or industry are more dominant than others. One of the very first examples was published by Baek et al. in 2005 [5]. Their analysis shows that the majority of companies in the stock market are influenced by energy companies. The authors also emphasized that transfer entropy is relatively better than other methods they used. In financial networks, entropy measures the uncertainty and exposes as a measure of randomness. For example, if stocks have high rate of entropy, they mey be riskier than others with lower value of entropy [14].

At the time of war, earthquake or illness, the stock market and the commodity market are the first institutions to be affected. Financial activity are subject to potential possible crisis. Studies have been carried out in order to predict the occurring potential crises. Gao and Hu have developed an early warning system for stocks [34]. The system may not warn significant financial quake but the system has well predicted the fluctuations in stocks prices. Their system is based on the Omori Law which is an early warning system for earthquake after stocks. They analysed the 2008 global financial crises and particularly the most attention-grabbing stocks like AIG and Lehman Brothers. They have demonstrate that the early warning system they have developed is really promising in predicting stock market crises. In fact, they showed that the system could also be generalized and applied to predict general economic recessions.

Another study on finance by Dimpfl and Peter allows determining, measuring and testing for information transfer without being restricted to linear dynamics [78]. Also, they examine the impact of the credit default swap market and the corporate bond market for the pricing of credit risk. Their analysis contains pre-crisis, crisis and post-crisis periods. The analysis shows the credit default swap market become more important during the crisis period.

In finance, according to the desired analysis, time series can be aggregated and analyzed over period ranging from one week to 10 years. In other words, a value can be created by analyzing the one-week value of a stock, or with its 10-year value. Marti et al. determined how many days are most suitable for clustering time series [60]. They analysed different types of models and clustering methods. The results showed time window varied between 250 to 500 realisations (roughly 1 to 2 years of daily returns) depending on the clustering methodology.

Harre focused on the 1997 Asian Financial Crises [41]. He measured the entropy, transfer entropy and Pearson correlation of the price change of the Dow Jones Industrial Average (DJIA) over the period of the Asian Financial Crisis. In the study on the comparison of transfer entropy and Pearson correlation methods, transfer entropy showed better results for detecting crises.

It shows that the Transfer Entropy has a wide application area with applications in social sciences, neuroscience, finance and applied physics. Whether it is just a statistical approach or a multidisciplinary approach like Omori law [34], it gives reliable results to the researchers.

Before giving the definition of the Transfer Entropy, there will be some definitions that we need to provide as prerequisites. We commence with mutual information.

Definition 15 (Mutual information)**.** *Mutual information (MI) measures the amount of information obtained about one random variable through knowledge of the other[72].*

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{2.5}$$

In terms of KLD,

$$I(X;Y) = D_{KL}(p(x,y) \parallel p(x)p(y)) \tag{2.6}$$

In terms of conditional and joint entropy,

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \end{aligned} \tag{2.7}$$

High mutual information indicates a large reduction in uncertainty, while a small value indicates a small reduction in uncertainty. If MI is zero, it means that the variables are independent. It can also be seen from Equation 2.5 that mutual information is symmetrical.

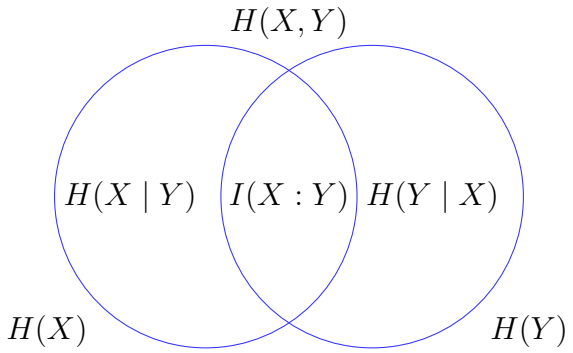

Figure 2.8: The relationship between entropy and mutual information expressed using a Venn diagram.

In Figure 2.8, $H(X)$ is represented by the entire set on the left, while $H(Y)$ is the entire set on the right. Mutual Information($I(X,Y)$) is the intersection of the left and right. Joint entropy ($H(X,Y)$) is the union of the two sets. Conditional entropies($H(X|Y)$,$H(Y|X)$) are above illustrated in the figure.

Mutual information is a measurement of the information that two sets share. It measures how much one the set tells us about another. Mutual information is a symmetrical quantity and it is not always easy to compute. There are some estimation methods for Mutual information which are conventional estimator binning methods, entropy estimators from k-nearest neighbour distance [46].

Mutual information does not convey dynamic or directed information. Time delay mutual information is suited for this purpose than standard mutual information. However, it is still not well suited to distinguish information. Schreiber proposed a new measurement referenced to as "Transfer Entropy", which is able to distinguish the causes and effects, and to detect asymmetry in the interaction of the component subsystems [69].

We can use the conditional mutual information to define the uncertainty of X due to knowledge of Y when Z is additionally given.

Definition 16 (Conditional Mutual Information)**.** *The conditional mutual information of random variables X and Y given Z is defined by [23, 24, 33, 42].*

$$\begin{aligned} I(X;Y|Z) &= H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z) \\ &= H(X \mid Z) + H(Y \mid Z) - H(X,Y \mid Z) \\ &= H(X \mid Z) - H(X \mid Y,Z) \end{aligned} \tag{2.8}$$

Conditional Mutual Information is also defined as,

$$I(X;Y|Z) = -\sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} p(x,y,z)\log_2 \frac{p(x,y,z),p(z)}{p(x,z)p(y,z)} \tag{2.9}$$

Transfer Entropy (TE) is a variant of Conditional Mutual Information which was first defined by Schreiber [69]

Definition 17 (Transfer Entropy)**.** *Transfer entropy is the conditional mutual information with the history of the influenced variable [42, 69],*

$$\begin{aligned} T_{Y\to X} &= I(X_t; Y_{t-1}|X_{t-1}) \\ &= H(X_t|X_{t-1}) - H(X_t|X_{t-1}, Y_{t-1}) \end{aligned} \tag{2.10}$$

Transfer Entropy is also defined as, [33, 49, 66]

$$
\begin{aligned}
T_{Y\to X} &= -\sum_{x\in X}\sum_{y\in Y} p(x_t, y_{t-1}, x_{t-1}) log_2 \frac{p(x_t|y_{t-1}, x_{t-1})}{p(x_t|x_{t-1})} \\
&= -\sum_{x\in X}\sum_{y\in Y} p(x_t, y_{t-1}, x_{t-1}) log_2 \frac{p(x_t, y_{t-1}, x_{t-1})p(x_{t-1})}{p(x_t, x_{t-1})p(y_{t-1}, x_{t-1})}
\end{aligned}
\tag{2.11}
$$

Here x_{t-1} and y_{t-1} are the past states of the x and y respectively, t is the time index.

After giving these definitions, we can illustrate them with the the following Venn diagram.

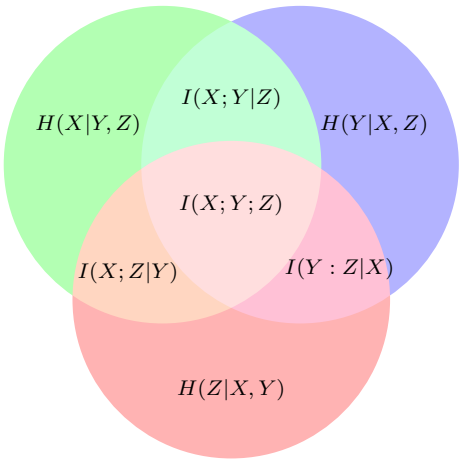

Figure 2.9: Various information measures associated with correlated variables X, Y and Z. In this Venn diagram $H(X|Y,Z)$, $H(Y|X,Z)$ and $H(Z|X,Y)$ are conditional entropies. $I(X;Y|Z)$, $I(X;Z|Y)$ and $I(Y;Z|X)$ show the conditional mutual information.

Transfer Entropy shows similarities with Granger Causality, which is another causality measure in the statistical analysing time series [38]. Barnett et al. showed that these two concepts are equal, when Gaussian variables are used [8].

We will continue by defining another method of measuring, which is very similar to Kullback–Leibler divergence. Jensen-Shannon deviation is a measure of the difference between probability distributions in terms of entropy difference associated with probability distributions [52].

Definition 18 (Jensen–Shannon Divergence (JSD))**.** *Suppose that P and Q are two probability distributions defined on the same probability space, then the*

Jensen–Shannon Divergence is defined as,

$$D_{JSD}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) \tag{2.12}$$

where $D_{KL}(P \parallel Q)$ *is the Kullback–Leibler divergence and* $M = \frac{1}{2}(P + Q)$.

The JSD defined in terms of entropy,

$$\begin{aligned} D_{JSD}(P \parallel Q) &= H(M) - \frac{1}{2}(H(P) + H(Q)) \\ &= H(P \oplus Q) - \frac{1}{2}(H(P) + H(Q)) \end{aligned} \tag{2.13}$$

where $H(P \oplus Q)$ *is the entropy associated with the corresponding probability distribution over of the union graph [6]. The union graph is called a graph that is the sum of the adjacency matrices of the two given graphs (P and Q in this example).*

The Jensen–Shannon divergence is a method of measuring the similarity between two probability distributions on the same probability space. Although JSD is based on the Kullback-Leibler divergence, JSD differs from KLD in some aspects such as its symmetry. It can be seen from Equations 2.12 and 2.13 that the JSD is a symmetrical measurement.

JSD is a positive-defined measurement. In the study of Lin, it is bounded by 1, given that one uses the base 2 logarithm [55].

$$0 \leq D_{JSD}(P \parallel Q) \leq 1$$

In general, the bound in base b is $log_b(2)$. More generally, the Jensen–Shannon divergence is bounded by $log_b(n)$ for more than two probability distributions [55],

$$0 \leq D_{JSD}(P_1, P_2, \cdots, P_n) \leq log_b(n)$$

The square root of the Jensen–Shannon divergence is a metric often referred to as Jensen–Shannon distance [31].

2.4 von Neumann Entropy

In quantum mechanical system, given the density matrix ρ, its von Neumann entropy is defined as

$$S(\rho) = -tr(\rho \ln\rho)$$

where tr denotes the trace operator. The von Neumann entropy of ρ can also be computed as the Shannon entropy of the spectrum of ρ

$$S(\rho) = -\sum_{i=1}^{|V|} \lambda_i \ln\lambda_i$$

where λ_i are the eigenvalues of the ρ, on condition that $0 \ln 0 = 0$. The von Neumann entropy was defined in quantum mechanics but, can be expressed in terms of the Shannon entropy associated with the eigenvalues of the density matrix.

2.4.1 von Neumann Entropy of Undirected Graphs

The spectral decomposition of The normalized Laplacian matrix of the graph G is defined as

$$\widetilde{L} = \sum_{i=1}^{|V|} \lambda_i \phi_i \phi_i^T$$

where λ_i are the eigenvalues and ϕ_i the corresponding eigenvectors of $\widetilde{L}$. With the spectrum of the normalized Laplacian matrix, we can take advantage of some useful spectral properties of the graph G. For example, the eigenvalues are bounded between 0 and 2 [17].

$$H_{vN} = -\sum_{i=1}^{|V|} \frac{\widetilde{\lambda_i}}{|V|} \ln \frac{\widetilde{\lambda_i}}{|V|}$$

where $|V|$ is the number of nodes in the graph.

2.4.2 Approximate von Neumann entropy

Han et.al. have shown how to approximate von Neumann entropy for undirected graph in terms of simple degree statistics using the quadratic approximation to the Shannon entropy $x \ln x \approx x(1-x)$ [40].

$$H_{VN} \approx 1 - \frac{1}{|V|} - \frac{1}{|V|^2} \sum_{(u,v)\in E} \frac{1}{d_u d_v}$$

This allows the efficient calculation for the network entropy in $O(N^2)$ rather than $O(N^3)$ from the normalised Laplacian spectrum [40].

2.5 Heat Diffusion on Graphs

The Heat diffusion models have been be widely applied in real world scenarios. Ma et al. proposed a model for social network marketing using Heat Diffusion on a graph [58]. Observing people's attention and shifting to certain topics on social media platforms such as Twitter can also be understood with a heat dissipation model. In another study, Thanou et al. used graph learning algorithm to detect patterns from Uber rides in New York City [77]. In this example the movement of people in buildings or vehicles in cities represented by a geographic information.

The heat kernel can also be thought of as describing the heat flow along the edges of the graph over time [37], where the flow rate is determined by the Laplacian of the graph. In particular, the graph Laplacian matrix is used to model the diffusion of heat along a graph. Bie et al. studied whether the trace of the heat kernel could be used to characterizing the properties of graphs [85]. We can give the heat equation associated with the Laplacian as follows,

$$\frac{\partial h_t}{\partial t} = -\widetilde{L} h_t \tag{2.14}$$

where $\widetilde{L}$ is normalised Laplacian matrix, h_t is the heat kernel and t is time. The heat kernel is the fundamental solution of the heat equation [85]. The solution to the heat equation is

$$h_t = e^{-t\widetilde{L}} \tag{2.15}$$

Exponentiating the Laplacian eigenspectrum will calculate the heat kernel on a graph [17].

$$\begin{aligned} h_t &= \Phi \exp[-\Lambda t]\Phi^T \\ &= \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i \phi_i^T \end{aligned} \tag{2.16}$$

The heat kernel is a $|V| \times |V|$ matrix. If we examine this equation element-wise for the G graph; Let u and v be two nodes of G,

$$h_t(u, v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{2.17}$$

Kondor and Lafferty focused on generating kernels on an undirected and unweighted graph [45]. One of their conclusions was that diffusion kernels can be practical use when standard sparse matrix techniques are used. They also stated that the key to the success of kernel-based algorithms is the implicit mapping from a data space to some, usually much higher dimensional, feature space that better captures the structure inherent in the data.

Chung et.al. studied a discrete version of heat kernel smoothing on graphs [20]. They demonstrated that the method can be used to smooth irregularly shapes on data in 3D images. They also developed an application that shows how to filter out the data in lung blood vessel trees obtained from tomography. This focuses on a weighted graph.

Chapter 3

Graph Time Series Analysis using Transfer Entropy

3.1 Introduction

Before we begin to explain transfer entropy, we need to explain the concept of entropy. The German physicist Rudolf Clausius introduced the concept of entropy in 1850 as a way of expressing the second law of thermodynamics. Clausius states that the total entropy of a closed system cannot decrease but will always increase [22]. Entropy is the measure of the disorder of a system and since the entropy will increase continuously in closed systems, the disorder is always in the direction of increase. However, within a closed system, the entropy of one system can decrease by increasing the entropy of another system [53]. But the total entropy of this system always tends to increase. For example, the growth of plants and trees in the world with the effect of sunlight reduces the current entropy, but the entropy of the solar system rises much more due to the solar flares that create these rays. In later years, the statistical definition of the concept of entropy was given by the Austrian physicist Ludwig Boltzmann [27]. The American scientist Willard Gibbs developed the interpretation of entropy in Statistical mechanics as a measure of uncertainty, disorder, or confusion [36]. In 1932, John von Neumann expanded the Gibbs entropy in classical statistical mechanics to quantum statistical mechanics [84]. This entropy has come to be called von Neumann entropy. This was then

followed by the application of entropy in probability theory by Claude Shannon (1948) [72] and with Shannon's approach, the foundations of information theory were laid.

Shannon introduced the concept of information entropy, which is a measurement of how much information there is in a signal. Shannon focused on how best to encode the information a sender sends to receiver, and he showed a method to measure a minimum number of bits required to send information without losing its meaning [72]. The method is the development of information theory and the measurement called "Shannon Entropy". This method is used in many algorithms and applications, from compression technologies [7] to information flow [3, 71] and fields of diverse as [9] and economics [2, 35]. For example, in finaince, entropy can be used to measures uncertainty and financial risk. If a stock has a high rate of entropy, it may be riskier than others.

3.2 Transfer Entropy

Schreiber developed the Transfer Entropy (TE) [69] concept based on Shannon entropy. It gives a time-asymmetric statistical measurement, which characterises the amount of information flow between two time series or from one variable to the another. So, TE shares some of the properties of mutual information but takes the dynamics of information into account, and it gives causal relations from which we can infer regulatory dynamics. Transfer entropy measures a directed relationship between variables. As seen in the Definition 17, $T_{Y\to X}$ and $T_{X\to Y}$ are not equal.

There are several estimation method for transfer entropy computation in the literature such as kernel estimation [65, 57, 90], the binning method [82, 24, 78], the k-nearest neighbor method [46, 89] and the symbolic method [59, 30, 74] among others. Transfer entropy can be useful in analysing complex systems such as biological or artificial systems [56, 83, 86, 88] or financial systems [41, 42, 59, 62, 88]. In general, there are two methods of establishing transfer entropy, which are mutual information and Kullback- Leibler Divergence (KLD). We will show them in the following sections.

Transfer Entropy as Conditional Mutual Information

Transfer Entropy $T_{Y\to X}$ which means information transition from Y to X, can be written as a Conditional Mutual Information. If we consider the Z in Equation 2.8 as the past value of X_t

$$T_{Y\to X} = I(X_{t+1}, Y_t|X_t) = H(X_{t+1}|X_t) - H(X_{t+1}|X_t, Y_t) \tag{3.1}$$

Here X and Y two processes, X_t and Y_t are the past states of the variable X and Y respectively. t is the time index.

Transfer Entropy as Kullback- Leibler Divergence

Transfer Entropy also can be explained as Kullback-Leibler Divergence.

$$T_{Y\to X} = D_{KL}(p(X_{t+1}, Y_t, X_t) \parallel p(X_{t+1}|X_t)p(Y_t|X_t)p(X_t)) \tag{3.2}$$

Here X and Y two processes, X_t and Y_t are the past states of the variable X and Y respectively. t is the time index. When calculating TE, Schreiber, himself used the KLD for conditional probabilities [69, 44].

Transfer Entropy on Edge Weighted Graphs

In this chapter we will use Schreiber's transfer entropy to develop a new entropic characterisation of graphs from time series data. We use the transfer entropy to weight the edges of a graph where the nodes represent time series data and the edges represent the degree of commonality of pairs of time series. The result is a weighted graph which captures th information transfer between nodes over specific time intervals. Then, the weighted normalised Laplacian were applied, which we defined in Definition 9. We characterise the network at each time interval using the von Neumann entropy computed from the normalised Laplacian spectrum, and study how this entropic characterisation evolves with time, and can be used to capture temporal changes in network structure.

Suppose that $G(V, E)$ is a graph with vertex set V and edge set $E \subseteq V \times V$.

Let u and v be any two elements of the node set V ($u, v \in V$). We use the transfer entropy to define an edge weight $W_{u,v}(t) = T_{u \to v}(t)$ between nodes u and v at time t. The thresholded weighted adjacency matrix A is defined as follows

$$A(u,v) = \begin{cases} W_{u,v}, & \text{if } W_{u,v} > threshold. \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Here we have applied a threshold to reduce noise in the system. The threshold we will choose here should be at a level that will both reduce the noise in the system and not affect the general characteristics of the system. Our method of determining the threshold was the trial and error method. As a result of the different threshold trials we made, by taking the strongest 5 percentage of edges. We both reduced the level of noise and, we have preserved the general characteristic of the system.

We have also constructed a graph from the clusters where the stocks are in the same sector to represent how the edge transfer entropy distributes itself across both within and between sector links. To do this suppose each node can be assigned a unique label μ_u and that these labels can be partitioned into a set of m sector-class-labels, $\Omega = \{\omega_1, ..., \omega_m\}$. In the case of the financial data analysed later in the paper, the node labels (μ_u) represent individual stock, while sector labels (ω_n) represent different commercial or industrial sectors to which individual stock belong. We can define a weighted sector neighborhood matrix with specified labels and elements.

$$AT_{\omega_a,\omega_b} = \sum_{\mu_u \in \omega_a} \sum_{\mu_v \in \omega_b} W_{u,v} \tag{3.4}$$

graph created on stocks in the same industry The graph created on stocks in the same industry $TG = (\Omega, AT)$ where Ω is the sector labels and AT the weighted adjacency matrix. The diagonal elements are the total transfer entropy associated with stocks within each sector, while off-diagonal elements are the total transfer entropy between pairs not belonging to the same sector.

For both graphs we need to compute the transfer entropy. To do this we

compute the normalised Laplacian matrix and from the eigenvalues of this matrix we compute the von Neumann entropy. The weighted degree matrix of graph G is a diagonal matrix D whose elements are given by $D(u,u) = d_u = \sum_{v \in V} A(u,v)$ The normalized Laplacian matrix of the graph G is defined as $\widetilde{L} = D^{-1/2}(D - A)D^{-1/2}$ and has elements

$$\widetilde{L} = \begin{cases} 1 & \text{if } u = v \text{ and } d_v \neq 0 \\ \frac{-1}{\sqrt{d_u d_v}} & \text{if } (u,v) \in E \\ 0 & otherwise \end{cases}$$

The spectral decomposition of the normalised Laplacian matrix is $\widetilde{L} = \sum_{i=1}^{|V|} \lambda_i \phi_i \phi_i^T$ where λ_i are the eigenvalues and ϕ_i the corresponding eigenvectors of $\widetilde{L}$.

3.2.1 Transfer Entropy for a graph

Suppose that an edge connects node u and node v, let these be any two elements of the node set V. That associated with the nodes are time series R_u and R_v. For each node of the time series is over a time window of duration Δt, and are denoted by $R_u(t) = \{u_{t-\Delta t}, u_{t-\Delta t+1}, \ldots, u_t\}$ and similarly $R_v(t) = \{v_{t-\Delta t}, v_{t-\Delta t+1}, \ldots, v_t\}$. To calculate the entropy transfer from node u to node v introduce a time delay (τ) for the windowed time series at node u, i.e. we consider the series $R_u(t+\tau) = \{u_{t+\tau-\Delta_\tau}, u_{t+\tau-\Delta_\tau+1}, \ldots, u_{t+\tau}\}$.

With these ingredients the entropy transfer is computable using $R_u(t)$, $R_v(t)$ and $R_u(t+\tau)$ [8, 57].

$$T_{u \to v}(t) = -\sum_t p(R_u(t+\tau), R_u(t), R_j(t)) \log_2 \frac{p(R_u(t+\tau)|R_u(t), R_v(t))}{p(R_u(t+\tau)|R_u(t))}$$

$$T_{u \to v}(t) = -\sum_t p(R_u(t+\tau), R_u(t), R_v(t)) \log_2 \frac{p(R_u(t+\tau), R_u(t), R_v(t)) p(R_u(t))}{p(R_u(t+\tau), R_u(t)) p(R_u(t), R_v(t))}$$

Edge Weighting via Time Series Cross-Correlation

Our aim is to explore which network characterisation allows for the most precise description of market crises. To this end, we construct representations based

on the time evolution of both edge-weighted (correlation-based and transfer-entropy-based) and unweighted market networks. We compute the Pearson Correlation coefficient between the node time series to compute an edge-weight. For nodes u and v the Pearson coefficient is

$$\rho(u,v) = Cov(R^u, R^v)/Var(R^u)Var(R^v)$$

where $Cov(R^u, R^v)$ is the covariance of the two time series and $Var(R^u)$ and $Var(R^v)$ are their individual variances. The edge weight is given by $W(u,v) = abs(\rho(u,v))$. The cross-correlation is calculated for all pairs of time series and gives a $V \times V$ cross-correlation matrix. The representation of similarity of pairs of graphs based on distribution of correlation coefficient, the edge (u,v) the probability is $p_{u,v} = W_{u,v}/\sum(W_{u,v})$. We convert the correlations to probabilities in order to compute Kullback-Leibler Divergence (KLD) between graphs.

3.3 Experiments

We will commence by giving informations about the compilation of data and it is preparation for analysis. Also same basic theoretical information about tools used will be given.

3.3.1 Data Collection

Knowing the data you use is one of the most crucial point in data analysis. We use real-time historical financial data in order to be able to visualise the data more clearly and to make reasonable assessments about the results. We chose 8 different sectors and 100 largest companies in each sector. So we have collected a total of 800 shares. We used `fetch` function in `MATLAB` to collect data from `Yahoo!` for each stock we chose earlier. Then, the dataset was cleaned to make it suitable for use. After this data clearing process, we have a total of 431 shares remaining. The reason why about 350 companies were eliminated is because the companies' IPO (Initial Public Offering) dates are after the start date of the dataset or the company closed between the dates of the dataset such as Facebook

or Lehman Brothers. To be more specific, Facebook's IPO was long after the first date of the dataset, while Lehman Brothers closed within the dataset dates. We tried to keep the time range of the stock prices as wide as possible and got around 5500 working days. The reason for not choosing a longer time is that as time increases, the number of potential stocks decreases. So if the number of days increased to 6000, the number of shares will decrease to 350. Our reference point is the last date of December 2016, because the date as close to the present day as possible. About 20 years of data will be sufficient for our study. We will be able to examine the stable time before the crises, the crisis times, and the post-crisis times. For this purpose, we have arranged the dataset to be around 5500 working days, from January 1995. There are various reference points in the dataset, such as the Financial crisis of 2007-2008, 2011 summer stock markets plunge, 2015 Chinese stock market turbulence, 2016 Brexit etc.

The log-returns of the closing prices are used for experiments over time, defined as

$$R_t = \ln(P_t) - \ln(P_{t-1})$$

at time t, where P_t and P_{t-1} are the closing prices at time t. The cross-correlation method to obtain a cross-correlation coefficient matrix for each 28 working day of period of time.

In the stock market, companies have certain ups and downs. These frequency vary according to what companies do and the industry they are in. For example, frequency of tech companies in an approximately 1-year (the ups and downs of stock prices depend on annual frequencies). This period is 3-5 years in automotive companies. The reason of this, people tend to change their phones almost every year, while they usually change their vehicles every 3-5 years. Instead of determining variable time windows for each stock, we tried to calculate by taking windows of 28 working days on average.

After all this data preparation, there are 431 labeled stocks emerged. These stocks were divided into 8 different sectors according to the business they serve. These sectors are *Basic Material, Consumer Goods, Financials, Healthcare, Industrial Goods, Services, Technology, Utilities.*

Basic Material	50 stocks
Consumer Goods	62 stocks
Financials	50 stocks
Healthcare	51 stocks
Industrial Goods	68 stocks
Services	49 stocks
Technology	44 stocks
Utilities	57 stocks
Total	431 stocks

Table 3.1: 8 different sectors and the number of stocks belonging to this sectors.

431 shares belonging to these sectors, and over a preiod of 5500 days. We obtain 5500 undirected date, 5500 directed data and 5500 correlation matrix.

3.3.2 Time Series Analysis using Transfer Entropy and von Neumann Entropy

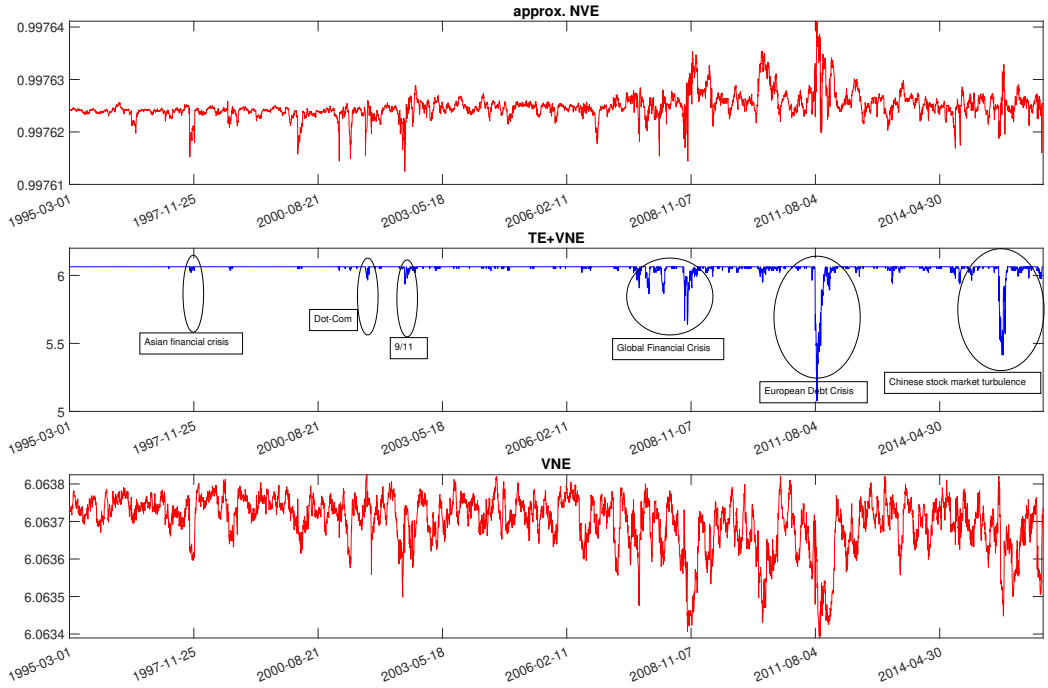

Figure 3.1: 3.1a (in red) is the von Neumann entropy computed from the normalised Laplacian spectrum. 3.1b (in blue) shows the transfer entropy applied of the weighted transfer entropy graph as a function of time. 3.1c (in red) is the approximate von Neumann entropy of Han et al [40].

In Figure 3.1 the main features to note are that the different financial crises

emerge more clearly when we use transfer entropy to weight the edges of the graph than when the two alternatives are used. From left to right the main peaks correspond to Asian financial crisis (1997), dot-com bubble (2000), 9/11 (2001), global financial crisis ($2007-08$), European debt crisis ($2009-12$), Chinese stock market turbulence ($2015-16$).

To take this analysis of the transfer entropy one step further we perform principal components analysis on a time series whose components are the total transfer entropies associated with each node in the graph.

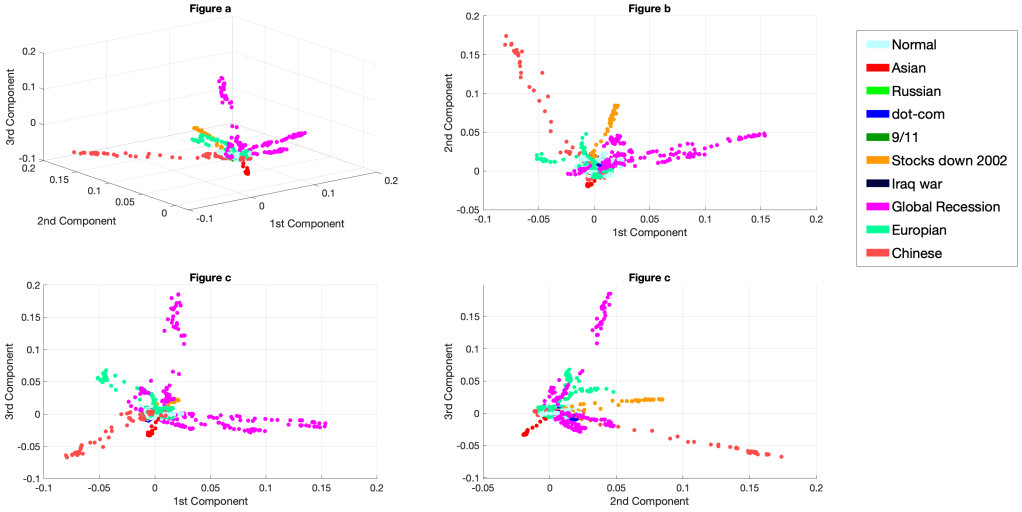

Figure 3.2: PCA for transfer entropy stock-price graphs. The Figure a shows the status of the first 3 components of PCA relative to each other. Figure b, Figure c and Figure d shows pairwise cases.

In Figure 3.2 we show different views of the leading three principal component projections of the time series. The different colours correspond to the financial epochs associated with different crises. It is interesting that the different crises correspond to different subspaces in the plot, following clearly clustered trajectories.

In Figure 3.3 we take this analysis one step further and show times series of the within and between sector transfer entropy for the finance and technology sectors. The financial sector dominates during the global financial crises when compared to other sectors. Moreover, the financial sector seems to be quite effective in determining the direction of the market during the crises in 2008. The technology

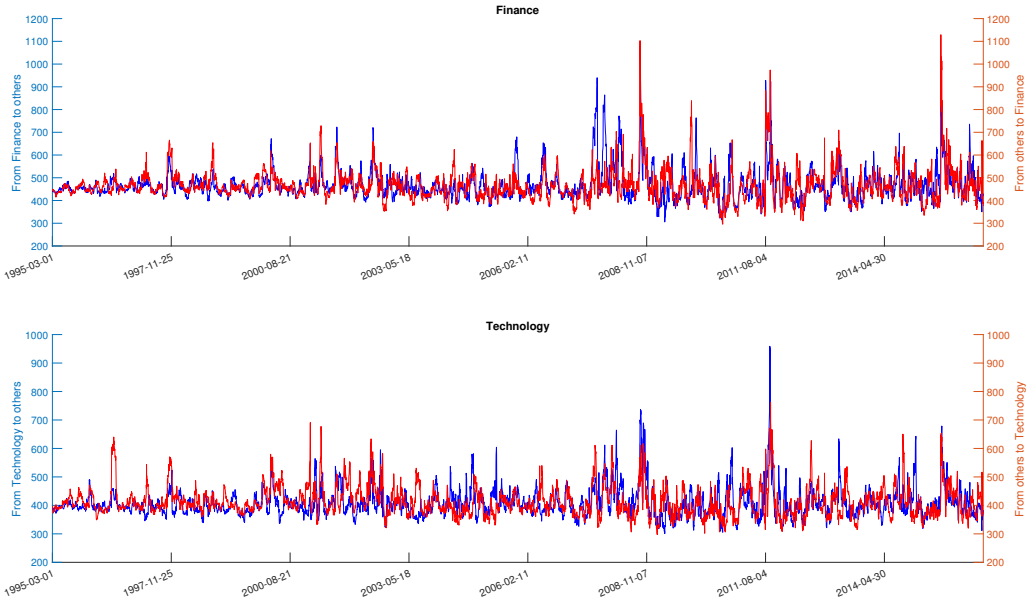

Figure 3.3: The top plot (Figure Finance) shows the information flow between the Finance sector and other stocks. The plot in blue in this figure shows the overall market impact of the finance sector over a period of time. The one in red shows the impact of the whole market on the financial sector over the same time period. The figure below (Figure Technology) shows the results of the same calculation for the technology sector

sector, on the other hand, is generally affected by the other sectors by the middle of the 2000's. After the Dot-com bubble, it gradually moves to a position that has affected the market. In the Europe and China financial crises, it has been observed to be passive.

3.3.3 Whole Network Visualization

In this section, we will visualize at the data we have created, using different statistical approaches. First, we will the sum of cross-correlation coefficient matrix at each time epoch, which allow let us visualise how all of a share-hold market moves in time. We will compare this data with S&P500's historical data of approximately 20 years.

As a reference, we will use the S&P500 index which is a stock market index based on the market capitalizations of 500 large companies listed on the NYSE or NASDAQ. We also use the largest companies in their sectors, so they can give information about each other in stock movement.

In addition, we have examined the entropy which is another analysis param-

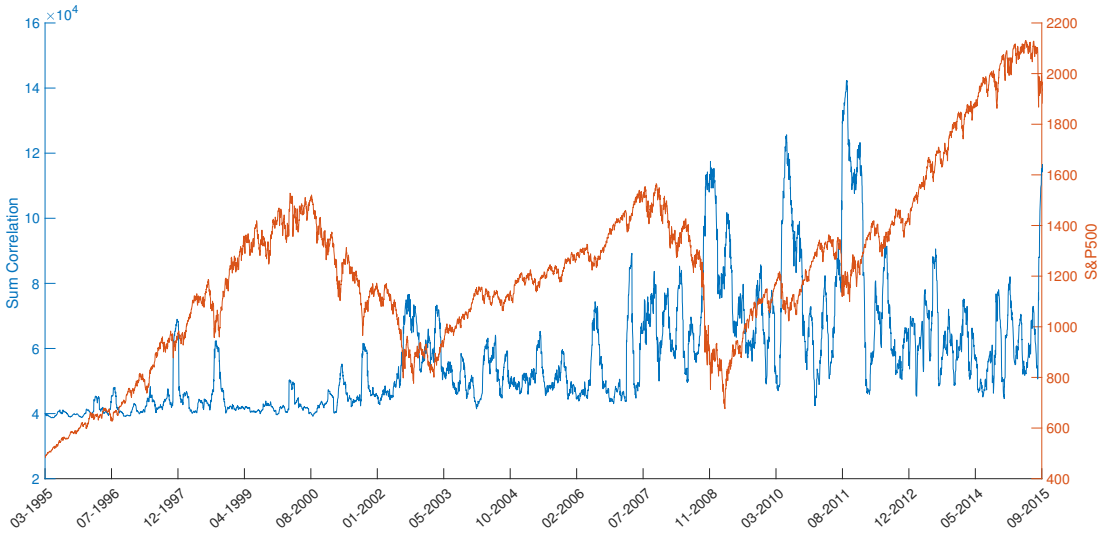

Figure 3.4: The blue line represents the sum of the corelation values at certain time epocs and the red line represents the S&P500 index. In November 2008 there is a 205% correlation increase against the S&P500 drop of 46%, in April 2010 there is a 257% correlation increase against the S&P500 drop of 15% and in August 2011 there is a 284% correlation increase against the S&P500 drop of 19%. The graph shows that while the market is moving downwards, the correlations have increased.

eter.

As in the Figure 3.5, there can be observed an inverse correlation between von Neumann Entropy and S&P 500 over the last few years. It is normal that the entropy of crisis times increases and firms try to get the best position to be influenced by the crisis.

3.3.4 Sectoral Network Visualization

We will evaluate 8 different sectors in this section and study their total corelations and entropies. The correlations of these sectors are calculated by the correlations of the stocks in each sector with each other. For example, when calculating the corelation for the Financial sector, only 50 stocks in the finance sector is computed. For other sectors it is calculated on this tab.

We will examine the comparison of the sectors with the general trend for about 20 years. We normalize the results (using the Euclidean normalisation, $||v|| = (\sum_k^N |v_k|^2)^{\frac{1}{2}}$) so that the data is more readable, then we look at the

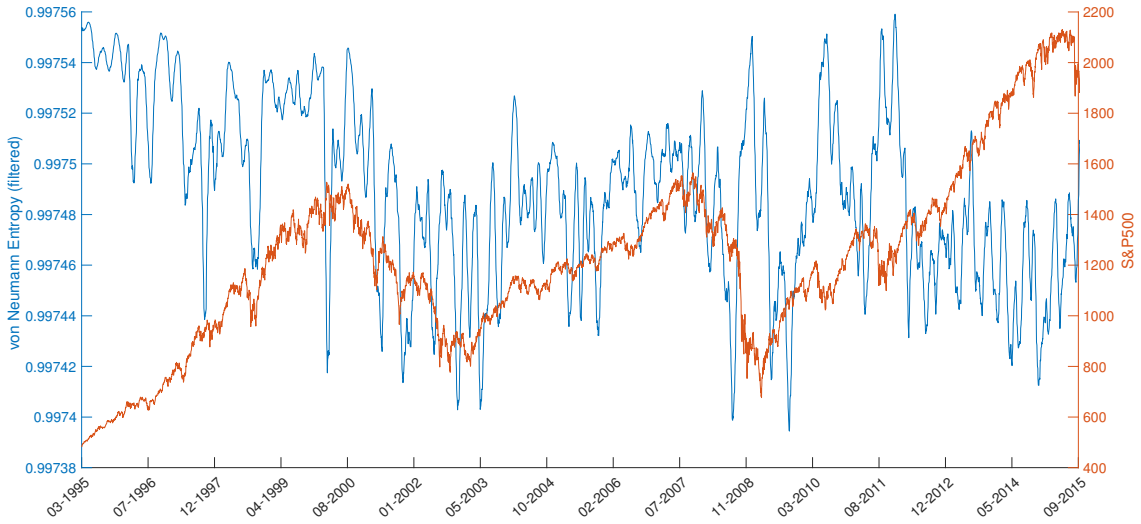

Figure 3.5: The blue line represents the von Neumann Entropy and the red line represents the S&P 500 index. We used a moving average filter for 20 days timewindow for the von Neumann Entropy. The figure shows that, in some cases there is a similar movement such around 1998, but it can be seen in 2001, 2009 or 2011 that von Neumann Entropy and S&P 500 are not in parallel.

correlation with total correlations.

We show the normalised sector connections in Figure 3.7. The Utilities show a much smaller correlation them to remaining sectors. On the other hand, with the new century, the trend of the Technology sector is becoming much more similar to the overall stock market. Around the 2008 crisis, we can observe that the Financial sector has disintegrated.

We are also see that Consumer Goods are almost always in perfect harmony with the general stock market trend. Also, Industrial Goods has a divergence from the general trend in the first decade. The interesting point is that unlike Figure 3.6, consumer goods have not seen a common move with the stock market until more recent years.

3.3.5 Non-Metric MultiDimensional scaling (Graph Clustering)

Non-Metric Multidimensional scaling (nmMDS) used the similarities (or dissimilarities) of a dataset. We can analyse any kind of similarity (or dissimilarity) matrix or correlation matrices using nmMDS [25, 48]. nmMDS which embed sets

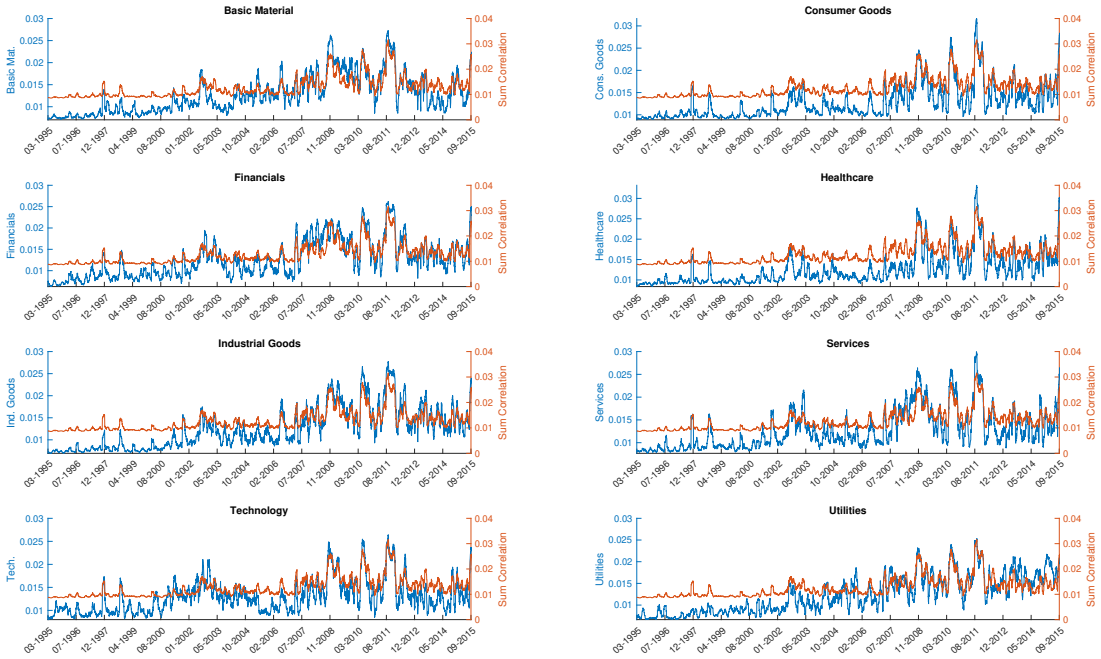

Figure 3.6: Correlations between sectors and all stock in our data (all 431 stocks without any sector separation). The sum of the intra-sector correlations and the correlations of all the stock are compared. Although almost all sectors showed a similar trend with the stock market, the Consumer Goods sector shows a very similar trend with the market in many crisis points.

of date defined in terms of similarity measurements rather than ordinal values into a vector space spanned by the eigenvectors of the similarity matrix [85]. If there is zero distance, maximum similarity is observed. The smaller the distance, the higher the similarity or the distance increases, the similarity decreases.

A simplified view of the nmMDS algorithm is as follows:

Step 1 - Assign nodes.

Step 2 - Compute the distances (D) among all pairs of nodes.

Step 3 - Calculate the stress function (shown in equation 3.5) according to the resultant distance (or correlation) matrix.

Step 4 - Adjust the coordinates of each point in the direction of stress.

Step 5 - Repeat steps 2-4 until the stress won't get any lower.

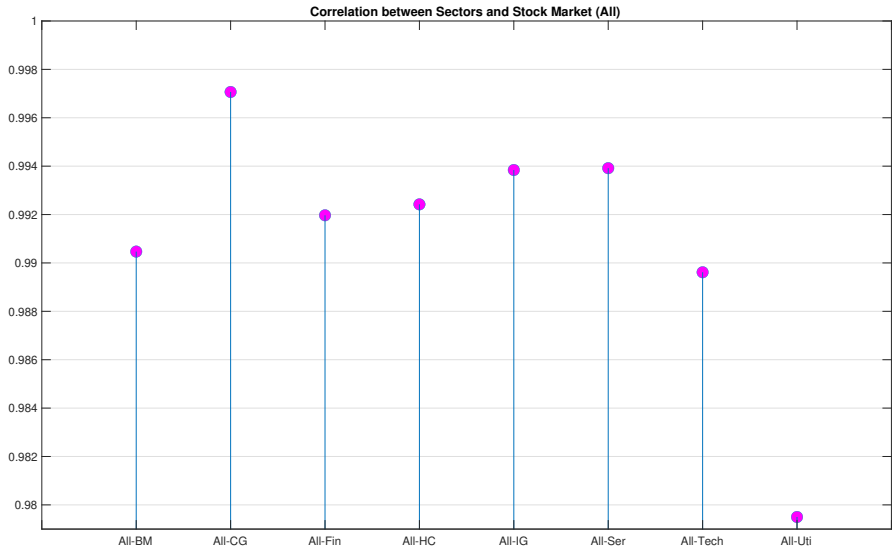

Figure 3.7: Correlation between sectors and the entire Stock Market. The mean total correlation values of the sectors are written as vectors and the correlation with whole markets is computed and compared. It is clear to see which sector is more correlated with general trend. The Consumer Goods which has been most compatible with the market for years. Differently, the Utilities was the sector that was at least in harmony with the total market

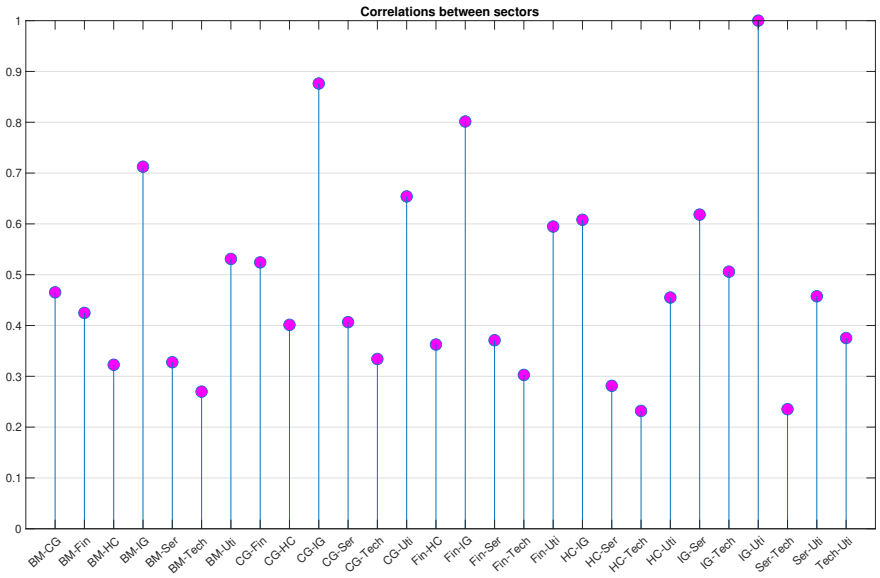

Figure 3.8: Correlation between sectors. The average total correlation values of the sectors are written as vectors and the correlation with each other is computed and compared. There is the highest correlation between Industrial Goods and Utilities. On average, Industrial Goodsn has the highest correlation between the other sectors.

Kruskal's non-metric MDS Stress function;

$$Stress = \sqrt{\frac{\sum(D-d)^2}{\sum(d^2)}} \tag{3.5}$$

the stress function, which is simply a measure of lack of fit between dissimilarities matrix (D) and adjusted distances (d) [12]. Kruskal [47] suggests the benchmarks in table 3.2.

Stress	Goodness-of-fit
0.200	poor
0.100	fair
0.050	good
0.025	excellent
0.000	perfect

Table 3.2: Stress Value Benchmark

In our own study we used MDS with cross-correlation. Which means, we used similarity values resulting from cross-correlation for each pair of nodes. The nodes in the MDS is already an assignment because it will hold 431x431 matrices and each matrix represents a specific day. Taking a 28 day time interval, 431 stocks were examined. 431x431 symmetric matrix appeared in correlation output. Then, after the 3rd step and afterwards. We tried to reach the result by minimizing the stress value. *mdscale* function are used on *Matlab* to do this calculations.

There is a fluctuation in S&P500 in between 2008 and 2012, as Figure 3.4 shows. To illustrate the results of MDS, we are dealing with the peak and bottom points of the fluctuation between these dates.

In Figure 3.9, as the average correlation values increase, in the general sense the sectors have established a much tighter connection with each other. Another point is that stocks of each sector are clustered. For example the clusters in the utilities sector are clearly visible in each graph. Another point that needs to be addressed is that the stress values are very small. If we take table 3.2 as reference, all stress values are good (around 0.05). We see that the stress value is slightly smaller 3-D space. In Figure 3.10, no clustering or pattern was observed. The stocks of the sectors are dispersed in both the 2d and 3d space. Stress values are

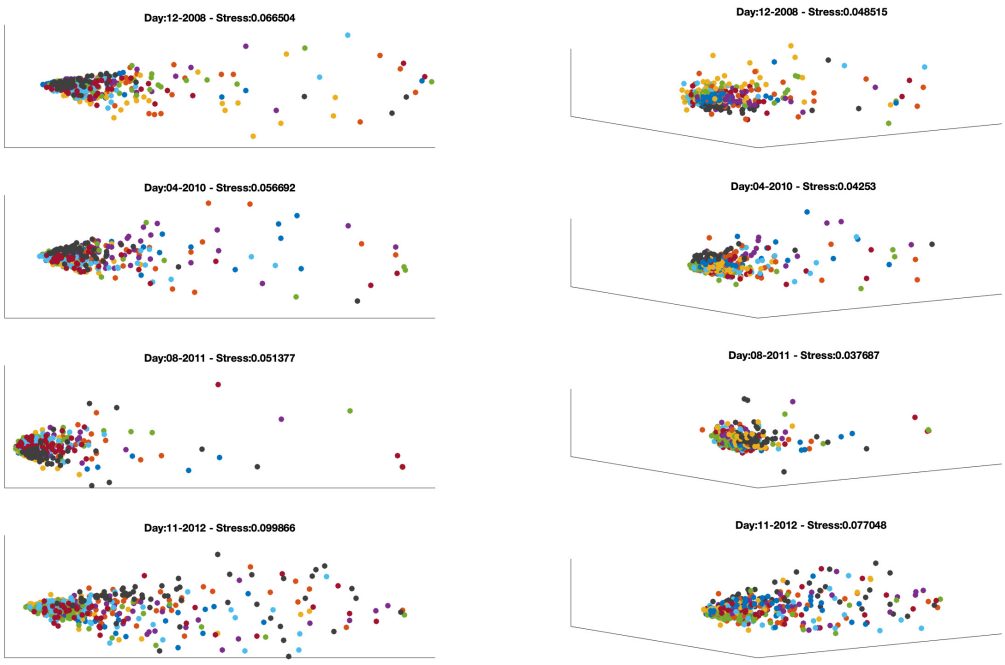

Figure 3.9: 2D and 3D MDS for peak points; each color represents different sectors, Basic Mat, Consumer Goods, Financials, Healthcare, Industrial Goods, Services, Technology, Utilities. On the selected dates, there is a regular, low entropy activity in the markets as the stress value is low.

above 0.1 levels in the 3D space, while in the 2D space this value reaches 0.25, and these values are at poor levels according to table 3.2.

3.4 Summary

We used transfer entropy to analyze a financial market dataset that includes closing prices of stocks traded over a 5400-day period. We commenced by constructing a graph in which the edges represent information flow between windowed time series for the stock, quantified using transfer entropy. It has been shown that the von Neumann entropy of the resulting weighted graph provides a better localization of temporal anomalies in the network structure due to global financial crises. Compared to the approximate von Neumann entropy, it is less prone to noise. Also, PCA of cumulative node transfer entropy over time shows that different financials occupy different subspaces that do not overlap substantially.

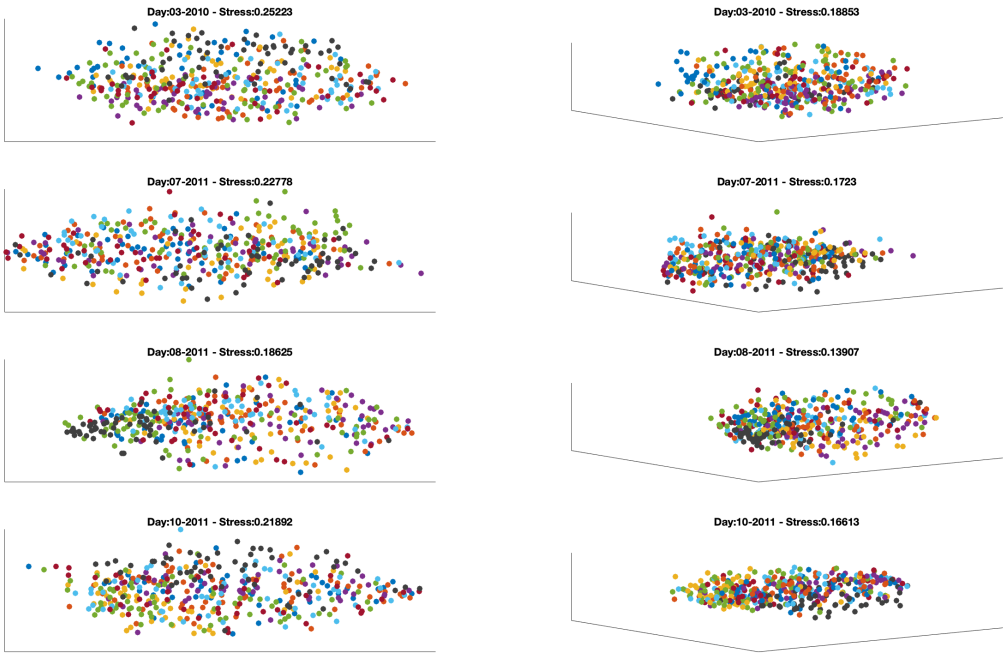

Figure 3.10: 2D and 3D MDS for Bottom points Basic Mat, Consumer Goods, Financials, Healthcare, Industrial Goods, Services, Technology, Utilities. Since the stress value is high on the selected dates (above), low entropy, a steady activity is not observed in the markets.

By reducing the dimensionality of the problem by considering a representation based on cumulative transfer entropy within and between sectors, we can still distinguish abnormal periods, but less clearly.

Thus, transfer entropy appeares to capture the flow of information within financial trading networks in a less noise-prone way than von Neumann entropy. However, this is at the expense of computational cost.

Chapter 4

Heat-Kernel Smoothing

The main purpose of smoothing is to minimize undesirable distortion and noise while maintaining important features. Diffusion-based filters have become a powerful and well-developed technique for smoothing [18, 19, 54]. Here we remove unwanted noise from both single graph and multigraph with heat kernel diffusion.

Kernel smoothing is a statistical method can be used to estimate a real valued function as the weighted average of nearby observed data. It expands on the notion of a moving average [18].

4.1 Heat Kernel on Graphs

Here we will use the Heat kernel as the kernel smoother. The heat diffusion on the graph $G = (V, E)$. The *Laplacian matrix* (L) of a graph (G) is a matrix that define as the difference between the degree matrix and the adjacency matrix. The normalized Laplacian matrix, defined as $\widetilde{L} = D^{-1/2}LD^{-1/2}$ [17].

Matrix form the heat equation on a graph associated with the Laplacian L is [17]

$$\frac{\partial H_t}{\partial t} = -LH_t \tag{4.1}$$

where the H_t is $|V|$x$|V|$ matrix and t is time. The heat flow along the edges of the graph with time, where the rate of flow is set by the Laplacian (L). The heat

kernel satisfies the initial condition $H_0 = I$, where I is the identity matrix. The solution of the heat equation is found by exponentiating the Laplacian matrix with time t,

$$\begin{aligned} H_t &= e^{-tL} \\ &= I - tL + \frac{t^2L^2}{2!} - \frac{t^3L^3}{3!} + \frac{t^4L^4}{4!} - \dots \end{aligned} \tag{4.2}$$

If we express Laplacian with eigenspectrum ($L = \Phi\Lambda\Phi$) and if we use in the equation above (equation 4.2). The heat kernel becomes

$$H_t = \Phi e^{-t\Lambda}\Phi$$

The heat kernel is a $|V|$x$|V|$ symmetric matrix with elements

$$H_t(i,j) = \sum_{v=1}^{|V|} \phi_v(i) e^{-t\lambda_v} \phi_v(j) \tag{4.3}$$

when t tends to zero, then $H_t \simeq I - L_t$, the heat kernel depends on the local connectivity structure of the graph. Else, if t is big, the kernel is governed by the global structure of the graph, here $H_t \simeq \phi_2 e^{-t\lambda_2}\phi_2^T$, where λ_2 is the smallest non-zero eigenvalue, i.e. the Fiedler vector [17, 21, 87].

$R(u)$ be noisy signal and $F(u)$ be the denoised signal at u-th vector for $u = 1, 2, \dots, |V|$, then we have

$$F_t(u) = \sum_{v=1}^{|V|} H_t(u,v) R(u) \tag{4.4}$$

where $H_t(u,v)$ denotes the (u,v) element of matrix H at time t [80].

4.2 Multilayer Network Structure

We will smooth the edge entropy by performing a diffusion operation on a Multilayer graph. Here, Multilayer graph will be explained and then directed causality will be defined on this graph.

4.2.1 Multilayer Graph

In basic network theory, a network is represented by a graph $G = (V, E)$, where V is the set of nodes and E the edges between nodes. This network will be referred to as a single layer graph. In Definition 1 we explained about single layer graph.

Here we use a similar formalism to characterize multilayer networks. If we assume that each different state the network is a different network, which in our study, these different states are created by states of the same network at different times.

A multilayer network can be represent by the set

$$\mathcal{G} = \{G^{[1]}, G^{[2]}, \ldots, G^{[M]}\}$$

where $G^{[n]}$ is a single layer graph at state n. Here $\mathcal{G}$ is a multilayer network with M layers [10, 64].

To calculate multilayer laplacian, also called supra-Laplacian, $A^{[n]}$ is adjacency matrix and $D^{[n]}$ is degree matrix of nth layer. Corresponding laplacian is $L^{[n]} = D^{[n]} - A^{[n]}$ for inter-layer network. The supra-Laplacian $\mathcal{L}$ of the whole multilayer may be separated in two contributions [39, 73]

$$\mathcal{L} = \mathcal{L}^N + \mathcal{L}^I \tag{4.5}$$

where $\mathcal{L}^N$ stands for the supra-Laplacian of the intralayers and $\mathcal{L}^I$ for the interlayer supra-Laplacian. The $\mathcal{L}^N$ is just the direct sum of the intralayer Laplacians,

$$\mathcal{L}^N = \begin{pmatrix} L^{[1]} & 0 & 0 & 0 \\ 0 & L^{[2]} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & L^{[n]} \end{pmatrix}$$

The interlayer supra-Laplacian ($\mathcal{L}^I$) may be expressed as the Kronecker prod-

uct of the interlayer Laplacian and the $N \times N$ identity matrix I [73, 26],

$$\mathcal{L}^I = L^I \otimes I$$

where L^I is interlayer Laplacian. $L^I = D - W^I$ where, $W^I \in N \times N$ whose components represent the adjacency of the connection between every pair of layers [73, 26]. Therefore,

$$\mathcal{L}\phi = (\mathcal{L}^N + \mathcal{L}^I)\phi = \lambda\phi \tag{4.6}$$

ϕ is an eigenvector of the full supra-Laplacian and λ is the eigenvalue corresponding to this eigenvector [73].

$$\mathcal{L} = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{pmatrix} + L^I \begin{pmatrix} I & -I & 0 \\ -I & I & -I \\ 0 & -I & I \end{pmatrix}$$

Here we represent a 3-layered Multilayer graph, where L_1, L_2, L_3 are the Laplacian matrices of the respective layers, and L^I is the interlayer diffusion coefficient.

4.3 Smoothing Edge Entropy

4.3.1 Graph Transformation

In our dataset, nodes represent a stock. The interaction of these stocks with each other represents the edges. We were able to quantify this interaction by calculating with Transfer Entropy (Section 3.2) and Cross-Correlation (section 3.2.1).

In order to define heat diffusion over edges, we need to convert the network from the node-based to edge-base. After the conversion, $G = (V, E)$ graph will turn into LG (Definition 4), which is $|V|^2 \times |V|^2$. Each edge from G become a node in LG. All weight of the edges of G become the node values in the LG. There are no edge weights in LG, all edge weights become 1. We will use the line graph to calculate heat diffusion on graph. After the diffusion process, we will make another conversion from the line graph to the original graph. So the

nodes in LG will transform into edges in G, along with this we will have made a diffusion for edges in G.

4.3.2 Heat Diffusion on Single Graph

A LG transformation will be made for each graph we have. Then we will have a new graph. Regardless of the interaction of this new graph with other new graphs, a diffusion process will be applied in the intralayer. Diffusion will have spread within the intralayer.

After this diffusion process, another transformation will be made and a new smooth G will be created from LG. The size of the newly created smooth G will also be in $V \times V$, like the original G. By calculating the total entropy of this newly created smooth G, we complete the smoothing process for a single-layer graph.

4.3.3 Heat Diffusion on Multilayer Graph

When doing heat diffusion for multilayer graphs, we start by doing a LG transformation for each graph, just like we do for single graphs. Heat diffusion will work slightly differently on multilayer graphs. In a multilayer graph, the heat transfer of a node is not only through the nodes of the layer to which it belongs. This transfer will also occur with the corresponding nodes in adjacent layers. The supra-laplacian calculation in equation 4.5 provides us this flux.

The diagonal part of the supra-laplacian represents the intra-layer flux, while the off-diagonal part represents the inter-layer flux. Since the heat flow is only between adjacent layers in inter-layer heat diffusion, the elements of the supra-laplacian matrix are zero except for the diagonal elements, one upper and one lower of the diagonals.

The size of each single-layer laplacian matrix inside the supra-laplacian (SL) matrix is $|V|^2$x$|V|^2$. When we construct a block diagonal matrix with these laplacians, the size of the resulting SL matrix becomes $|V|^2M \times |V|^2M$, where M indicates the number of layers.

Gomez et al. have shown that spectral decomposition, which we can divide

the matrix into eigenvalue and eigenvector, can be applied for SL [39]. Thus, a heat diffusion is applied to the entire multilayer network as in equation 4.3. After diffusion, we do one more transformation from LG to G. Finally, we look at the total entropy of each layer in the multilayer graph [28]. The entropy value is calculated separately for each graph. Another purpose of the second transformation is to make a more efficient analysis for labeled data sets. A comparison can be made with the node in the original network and the node created after diffusion.

4.4 Experiments

In this section, we will see the applications related to smoothing that we have done on graphs. As I explained in the section 3.3.1, I will use my own data set that I have created. The elements of this dataset are stocks. I choose this data set both because I have used it in previous experiments and because I know its labels. Thus, it was possible for me to make comments about the labels when needed.

Figure 4.1 shows the heat diffusion value applied for different time values. Here, heat diffusion is applied to each layer individually. Afterwards, the total entropy value within each layer is calculated. From a data set of about 5500 layers, the same amount of total entropy value is formed. Here, there was no smoothing at t=0. For the t=0.05 and t=0.1, the noise disappeared significantly, in addition, the spikes showing the characteristics of the plot were still clearly visible. For the t=0.2, spikes began to disappear together with the noise.

Figure 4.2 shows the 5-Layered Multilayer Graph Heat kernel Smoothing with different time values.

Here, heat diffusion is calculated on Multi Graphs created with 5 graphs

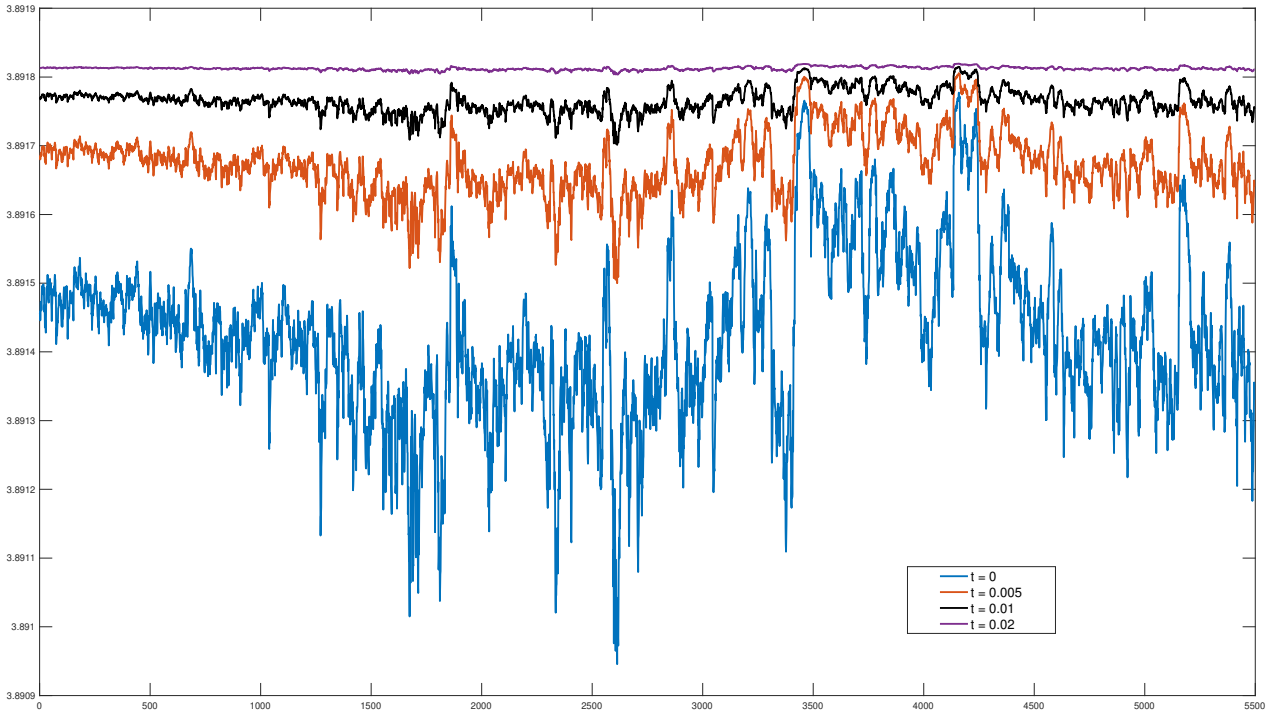

Figure 4.1: Single Graph Heat kernel Smoothing with different time values. Blue indicates t=0, that is, the absence of any smoothing. In the graph, the t value gradually increased and finally the purple graph was drawn for the t=0.02 value. Here it can be seen that as the t value increases, the smoothing increases. At the t=0.02 level, the spikes are close to disappearing.

selected sequentially.

$$\mathcal{L} = \begin{pmatrix} L_1 & 0 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 & 0 \\ 0 & 0 & L_3 & 0 & 0 \\ 0 & 0 & 0 & L_4 & 0 \\ 0 & 0 & 0 & 0 & L_5 \end{pmatrix} + L^I \begin{pmatrix} I & -I & 0 & 0 & 0 \\ -I & I & -I & 0 & 0 \\ 0 & -I & I & -I & 0 \\ 0 & 0 & -I & I & -I \\ 0 & 0 & 0 & -I & I \end{pmatrix}$$

Here we represent a 5-layered Multilayer graph, where L_1, L_2, L_3, L_4, L_5 are the Laplacian matrices of the respective layers, and L^I is the interlayer diffusion coefficient.

Here, heat diffusion is applied to 5-layered graph one by one. Afterwards, the total entropy value within each layer is calculated. Although our dataset has

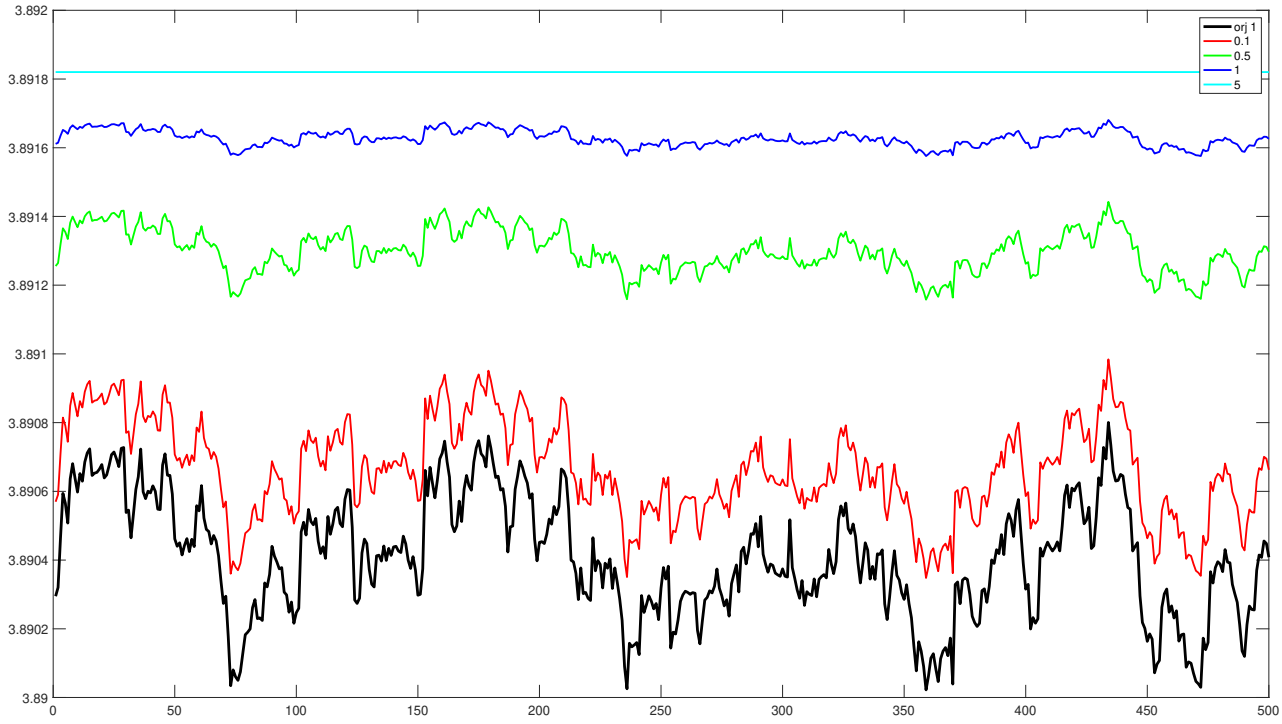

Figure 4.2: 5-Layered Multilayer Graph Heat kernel Smoothing with different time values. Black indicates t=0, it shows the original graph without any smoothing. The t value was gradually increased in the graph and finally a turquoise graph was drawn for the t=5 value. As the t value increased, although the distance between the peak point and the negative peak decreased, no significant noise reduction was observed.

5500 layers, we tried to make a smoothing using only 5 layers. The biggest reason for this is the converted LG data set that comes in very large sizes. Even only 5-layered graph has 928805×928805 size. As can be seen from the Figure 4.2 , this smoothing was only possible for 500 different layers. The reason for this is that the calculations of the resulting matrices take a very long time.

4.5 Summary

Here we demonstrate a graphical signal noise reduction method based on heat kernel smoothing. Experimental results demonstrated the effectiveness of the proposed methods. However, only heat kernel smoothing has been studied here. It will be interesting to examine how it works with other smoothing methods.

Basically, the aim of this study was to achieve noise reduction by performing heat diffusion on a multilayer graph. However, in the later stages of the study, the experiments could not be carried out as predicted due to the high computation cost. Even so, promising results were obtained in the method created by single layer smoothing.

The t value in heat diffusion has a very significant value in noise reduction. When this t value is selected big, the peak points in the graph begin to fade as expected. On the contrary, when t value is selected too small, the smothing will be so minimal that it cannot be observed,

Chapter 5

Conclusion

5.1 Contribution

We used transfer entropy to analyze data on closing prices by considering different financial datasets. The datasets are reliable enough in terms of the number of stocks as well as the number of days covered. The data set contains almost all of the major crisis events in the last twenty years. These stocks are selected with the largest market value. The biggest reason for this big companies plays a big role in market movements by their Market Cap and inspiring small companies which has small Market Cap and listed on the stock market. We are constructing a graph in which the edges represent information flow between time series for a stock, quantified using transfer entropy. The von Neumann entropy of the resulting weighted graph has been shown to provide better localisation of temporal anomalies in network structure due to global financial crises. Transfer entropy is less prone to noise compared to the approximate von Neumann entropy of Han et. al. [40], described as Chapter 3.

Reducing the dimensionality of the problem we can still separate anomalous epochs, but less clearly. So, considering the cost of this calculation, we can say that, transfer entropy appears to capture information flow within the financial trading networks in a manner which is less prone to noise than von Neumann entropy.

As another outcome, we have analysed how transfer entropy and cross-correlation

can be used to construct both unweighted and weighted network representations of time-evolving data. We explore how the entropy of these graphs, both undirected and directed, can be used to detect network anomalies. The result is that transfer entropy performs better than time-series cross-correlation when our dataset is considered. Moreover, if unweighted networks are used, the characterisations are more stable, and these two measures are used to threshold edges instead of weighting them.

Another result, we extracted from the heat diffusion study for Multiplayer graphs. A significant noise reduction is achieved by heat diffusion in single layer graphs. But here it is necessary to choose the t value wisely. In this study, a smoothing was also attempted for Multilayer graph, but it is far from the expected result.

5.2 Limitations

There are a number of limitations with the methods proposed. First, it should be noted that cross-correlation analysis can only be applied to measure the pairwise correlation between time series. A good comparison may not be accepted because it cannot give information about the causal relationship between time series. However, transfer entropy gives a causal relationship. The results may differ according to the approach methods used in calculating the TE. Although we use the binning method in terms of computational cost here, the results of different approach methods can also be analyzed,

A limitation can be mentioned for the causal relationship for the transfer entropy. The causal relationship between a pair of time-series could be directed, or indirect, which means a third time-series may be an agent for the causality or just a combination of both. If there were a third time-series in effect, transfer entropy could not determine it.

In this study, we encountered the biggest limitation in multilayer graph smoothing analysis. Here, we had very serious computation limitations both in transformation (G to LG and LG to G) and in the calculation of heat diffusion after supra laplacian. The size of the resulting matrix was very large, and it was quite

challenging for a 5500 layer data set, even for a 5 layer data set.

5.3 Future Work

Our future work will focus on how to use the transfer entropy representation to construct kernel representations of graph time series. Also, one can look at the results of the different methods used to calculate TE.

It is possible to try to explore a local structure rather than a global style adopted in the thesis. Also, we aim to explore how to transfer entropy can be used to analyse single networks, and that clustering node and explore node salience using centrality and related measures.

Researching machine learning applications with transfer entropy that will detect anomalies in advance seems to be promissing. We plan to make an ML application that will detect anomalies in advance.

Heat diffusion for multilayer graph also needs new studies to minimize computational cost.

Bibliography

[1] Laplacian eigenmaps and spectral techniques for embedding and clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. The MIT Press.

[2] *The Entropy Theory of Value*, pages 16–31. 2005.

[3] Nihat Ay and Daniel Polani. INFORMATION FLOWS IN CAUSAL NETWORKS. 11(1):17–41.

[4] N. Ahmad Aziz. Transfer entropy as a tool for inferring causality from observational studies in epidemiology. 06 2017.

[5] S. K. Baek, W.-S. Jung, O Kwon, and H.-T. Moon. Transfer Entropy Analysis of the Stock Market. *ArXiv Physics e-prints*, 2005.

[6] L Bai, E R Hancock, and P Ren. Jensen-Shannon graph kernel using information functionals. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2877–2880, 2012.

[7] Kedarnath J. Balakrishnan and Nur A. Touba. Relationship between entropy and test data compression. 26(2):386–395.

[8] Lionel Barnett, Adam B Barrett, and Anil K Seth. Granger Causality and Transfer Entropy Are Equivalent for Gaussian Variables. *Phys. Rev. Lett.*, 103(23):238701, 2009.

[9] Mridula Batra and Rashmi Agrawal. Comparative analysis of decision tree algorithms. In Bijaya Ketan Panigrahi, M. N. Hoda, Vinod Sharma, and

Shivendra Goel, editors, *Nature Inspired Computing*, pages 31–36, Singapore, 2018. Springer Singapore.

[10] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. 89(3):032804.

[11] Peter Bloomfield. *Fourier analysis of time series: an introduction*. Wiley series in probability and mathematical statistics. Wiley.

[12] Andreas Buja, Deborah F Swayne, Michael L Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data visualization with multidimensional scaling. 17(2):444–472.

[13] Ibrahim Caglar and Edwin R. Hancock. Graph time series analysis using transfer entropy. In Xiao Bai, Edwin R. Hancock, Tin Kam Ho, Richard C. Wilson, Battista Biggio, and Antonio Robles-Kelly, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 11004, pages 217–226. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

[14] Ibrahim Caglar and Edwin R. Hancock. Network time series analysis using transfer entropy. In Donatello Conte, Jean-Yves Ramel, and Pasquale Foggia, editors, *Graph-Based Representations in Pattern Recognition*, volume 11510, pages 194–203. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

[15] Guido Caldarelli, Stefano Battiston, Diego Garlaschelli, and Michele Catanzaro. Emergence of Complexity in Financial Networks. In Eli Ben-Naim, Hans Frauenfelder, and Zoltan Toroczkai, editors, *Complex Networks*, pages 399–423. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[16] Eric Chaisson. Energy flows in low-entropy complex systems. 17(12):8007–8018.

[17] F.R.K. Chung. *Spectral Graph Theory*. Number no. 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences.

[18] Moo K. Chung. Gaussian kernel smoothing.

[19] Moo K. Chung, Steven M. Robbins, Kim M. Dalton, Richard J. Davidson, Andrew L. Alexander, and Alan C. Evans. Cortical thickness analysis in autism with heat kernel smoothing. 25(4):1256–1265.

[20] Moo K. Chung, Yanli Wang, and Guorong Wu. Discrete Heat Kernel Smoothing in Irregular Image Domains. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5101–5104, Honolulu, HI, July 2018. IEEE.

[21] Moo K. Chung, Yanli Wang, and Gurong Wu. Heat Kernel Smoothing in Irregular Image Domains. *arXiv:1710.07849 [cs, eess, stat]*, October 2017.

[22] R. Clausius. Ueber verschiedene für die anwendung bequeme formen der hauptgleichungen der mechanischen wärmetheorie. *Annalen der Physik*, 201(7):353–400, 1865.

[23] T. M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley series in telecommunications. Wiley, New York, 1991.

[24] Thomas M Cover and Joy A Thomas. Entropy, Relative Entropy and Mutual Information. In *Elements of Information Theory*, pages 12–49. John Wiley & Sons, Inc., 2001.

[25] Michael A. A. Cox and Trevor F. Cox. *Multidimensional Scaling*, pages 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[26] Emanuele Cozzo, Guilherme Ferraz de Arruda, Francisco A. Rodrigues, and Yamir Moreno. Multilayer networks: Metrics and spectral properties. In Antonios Garas, editor, *Interconnected Networks*, pages 17–35. Springer International Publishing. Series Title: Understanding Complex Systems.

[27] Olivier Darrigol. 420Lectures on Gas Theory (1896, 1898). In *Atoms, Mechanics, and Probability: Ludwig Boltzmann's Statistico-Mechanical Writings - An Exegesis*. Oxford University Press, 02 2018.

[28] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. 6(1):6864.

[29] Lloyd Demetrius and Thomas Manke. Robustness and network evolution—an entropic principle. *Physica A: Statistical Mechanics and its Applications*, 346(3–4):682–696, 2005.

[30] Henning Dickten and Klaus Lehnertz. Identifying delayed directional couplings with symbolic transfer entropy. *Phys. Rev. E*, 90(6):62706, 2014.

[31] D M Endres and J E Schindelin. *A new metric for probability distributions*, volume 49. 2003.

[32] Hamza Farooq, Yongxin Chen, Tryphon T. Georgiou, Allen Tannenbaum, and Christophe Lenglet. Network curvature as a hallmark of brain structural connectivity. 10(1):4937, 2019-12.

[33] Stefan Frenzel and Bernd Pompe. Partial Mutual Information for Coupling Analysis of Multivariate Time Series. *Physical Review Letters*, 99(20):204101, 2007.

[34] Jianbo Gao and Jing Hu. Financial crisis, Omori's law, and negative entropy flow. *International Review of Financial Analysis*, 33:79–86, 2014.

[35] N.R. Georgescu and N. Georgescu-Roegen. *The Entropy Law and the Economic Process*. A Harvard paperback: Economics. Harvard University Press, 1971.

[36] J. Willard Gibbs. *A method of geometrical representation of the thermodynamic properties of substances by means of surfaces*. New Haven, 1873.

[37] P.B. Gilkey. *Invariance Theory: The Heat Equation and the Atiyah-Singer Index Theorem*. Studies in Advanced Mathematics. CRC Press, 2018.

[38] C W J Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424–438, 1969.

[39] S. Gómez, A. Díaz-Guilera, J. Gómez-Gardeñes, C. J. Pérez-Vicente, Y. Moreno, and A. Arenas. Diffusion Dynamics on Multiplex Networks. *Phys. Rev. Lett.*, 110(2):028701, January 2013.

[40] Lin Han, Francisco Escolano, Edwin R Hancock, and Richard C Wilson. Graph characterizations from von Neumann entropy. *Pattern Recognition Letters*, 33(15):1958–1967, 2012.

[41] Michael Harré. Entropy and Transfer Entropy: The Dow Jones and the Build Up to the 1997 Asian Crisis. In Hideki Takayasu, Nobuyasu Ito, Itsuki Noda, and Misako Takayasu, editors, *Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014*, pages 15–25. Springer International Publishing, Cham, 2015.

[42] Katerina Hlaváčková-Schindler, Milan Paluš, Martin Vejmelka, and Joydeep Bhattacharya. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46, 2007.

[43] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[44] A Kaiser and T Schreiber. Information transfer in continuous processes. *Physica D: Nonlinear Phenomena*, 166(1–2):43–62, 2002.

[45] Risi Kondor and John D. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *ICML*, pages 315–322, 2002.

[46] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):66138, 2004.

[47] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. 29(1):1–27.

[48] Joseph Kruskal and Myron Wish. *Multidimensional Scaling*.

[49] O Kwon and J.-S. Yang. Information flow between stock indices. *EPL (Europhysics Letters)*, 82(6):68003, 2008.

[50] Okyu Kwon and Gabjin Oh. Asymmetric information flow between market index and individual stocks in several stock markets. *EPL (Europhysics Letters)*, 97(2):28007, jan 2012.

[51] Lucas Lacasa and Ryan Flanagan. Time reversibility from visibility graphs of nonstationary processes. *Phys. Rev. E*, 92(2):22817, 2015.

[52] P. W. Lamberti, A. P. Majtey, A. Borras, M. Casas, and A. Plastino. Metric character of the quantum Jensen-Shannon divergence. *Phys. Rev. A*, 77(5):052311, May 2008. Publisher: American Physical Society.

[53] P.T. Landsberg. Can entropy and "order" increase together? *Physics Letters A*, 102(4):171–173, 1984.

[54] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex Networks: Principles, Methods and Applications*. Cambridge University Press, 2017.

[55] Jianhua Lin. Divergence measures based on the shannon entropy. page 7.

[56] Michael Lindner, Raul Vicente, Viola Priesemann, and Michael Wibral. TRENTOOL: A Matlab open source toolbox to analyse information flow in time series data with transfer entropy. *BMC Neuroscience*, 12(1):1–22, 2011.

[57] Joseph Troy Lizier. JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI*, 1, 2014.

[58] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, page 233. ACM Press.

[59] R. Marschinski and H. Kantz. Analysing the information flow between financial time series. *European Physical Journal B*, 30(2):275–281, 2002.

[60] Gautier Marti, Sébastien Andler, Frank Nielsen, and Philippe Donnat. Clustering Financial Time Series: How Long is Enough? 1603, 2016.

[61] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard business review*, 90(10):60–68, 2012.

[62] Anna Nagurney. Networks in Finance. In Detlef Seese, Christof Weinhardt, and Frank Schlottmann, editors, *Handbook on Information Technology in Finance*, pages 383–419. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[63] M E J Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

[64] V. Nicosia, G. Bianconi, V. Latora, and M. Barthelemy. Growing multiplex networks. 111(5):058701.

[65] Jan Orava. *K-nearest neighbour kernel density estimation, the choice of optimal k*, volume 50. 2011.

[66] Mikhail Prokopenko and Joseph T Lizier. Transfer Entropy and Transient Limits of Computation. *Scientific Reports*, 4:5394, 2014.

[67] Dymitr Ruta. Automated trading with machine learning on big data. In *2014 IEEE International Congress on Big Data*, pages 824–830. IEEE.

[68] Romeil Sandhu, Tryphon Georgiou, Ed Reznik, Liangjia Zhu, Ivan Kolesov, Yasin Senbabaoglu, and Allen Tannenbaum. Graph Curvature for Differentiating Cancer Networks. *Scientific Reports*, 5:12323, 2015.

[69] Thomas Schreiber. Measuring Information Transfer. *Physical Review Letters*, 85(2):461–464, 2000.

[70] Ahmet Sensoy, Cihat Sobaci, Sadri Sensoy, and Fatih Alali. Effective transfer entropy approach to information flow between exchange rates and stock markets. 68:180–185.

[71] Nikhil Seshadri and Michael Galperin. Entropy and information flow in quantum systems strongly coupled to baths. *Phys. Rev. B*, 103:085415, Feb 2021.

[72] C E Shannon. A mathematical theory of communication. *Bell System Technical Journal, The*, 27(3):379–423, 1948.

[73] Albert Sole-Ribalta, Manlio De Domenico, Nikos E. Kouvaris, Albert Diaz-Guilera, Sergio Gomez, and Alex Arenas. Spectral properties of the laplacian of multiplex networks. 88(3):032807.

[74] Matthäus Staniek and Klaus Lehnertz. Symbolic Transfer Entropy. *Physical Review Letters*, 100(15):158101, 2008.

[75] J. J. Sylvester. On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, with three appendices. *American Journal of Mathematics*, 1(1):64–104, 1878.

[76] Allen Tannenbaum, Chris Sander, Liangjia Zhu, Romeil Sandhu, Ivan Kolesov, Eduard Reznik, Yasin Senbabaoglu, and Tryphon Georgiou. Graph Curvature and the Robustness of Cancer Networks. 1502, 2015.

[77] Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard. Learning Heat Diffusion Graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):484–499, September 2017.

[78] Dimpfl Thomas and Peter Franziska Julia. Using transfer entropy to measure information flows between financial markets. *Studies in Nonlinear Dynamics & Econometrics*, 17(1):85–102, 2013.

[79] Clifford Ambrose Truesdell. Act IV. Internal Energy: the First Paper of Clausius. Entropy: the First Paper of Rankine. In M. J. Klein and G. J. Toomer, editors, *The Tragicomical History of Thermodynamics, 1822–1854*, volume 4, pages 187–218. Springer New York, New York, NY, 1980. Series Title: Studies in the History of Mathematics and Physical Sciences.

[80] Chien-Cheng Tseng and Su-Ling Lee. Distributed implementation of heat kernel smoothing for graph signal denoising. In *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4. IEEE.

[81] J Veiga, R C P Faria, G P Esteves, A J Lopes, J M Jansen, and P L Melo. Approximate entropy as a measure of the airflow pattern complexity in asthma. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 2463–2466. IEEE.

[82] Greg Ver Steeg and Aram Galstyan. Information transfer in social media. *Proceedings of the 21st international conference on World Wide Web (WWW '12)*, pages 509–518, 2012.

[83] Jonathan D Victor. Binless strategies for estimation of information from neural data. *Physical Review E*, 66(5):51903, 2002.

[84] John VonNeumann. *Mathematische Grundlagen der Quantenmechanik*. Springer, 1932.

[85] Bai Xiao, Edwin R. Hancock, and Richard C. Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, November 2009.

[86] Kai Yu, Liang Ji, and Xuegong Zhang. Kernel Nearest-Neighbor Algorithm. *Neural Processing Letters*, 15(2):147–156, 2002.

[87] Fan Zhang and Edwin R. Hancock. Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, 41(11):3328–3342, November 2008.

[88] Rongxi Zhou, Ru Cai, and Guanqun Tong. Applications of Entropy in Finance: A Review. *Entropy*, 15(11):4909, 2013.

[89] Jie Zhu, Jean-Jacques Bellanger, Huazhong Shu, and Régine Le Bouquin Jeannès. Contribution to Transfer Entropy Estimation via the k-Nearest-Neighbors Approach. *Entropy*, 17(6):4173, 2015.

[90] K Zuo, J J Bellanger, C Yang, H Shu, R Le Bouquin Jeann, and x00E. Exploring neural directed interactions with transfer entropy based on an adaptive kernel density estimator. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4342–4345.