Imperial College London

Final Year Project Report

Community Detection in Networks

Author:
Hesam Ipakchi

Supervisor:
Dr. Moez Draief
Second marker:
Dr. Jeremy Pitt

This report is submitted in fulfilment of the requirements
for the degree of *MEng Electronic and Information Engineering*
in the
Department of Electrical and Electronic Engineering
Imperial College London

March 2014

Abstract

Write Abstract here...

Acknowledgements

Write Acknowledgements here...

Contents

List of Figures

List of Tables

Notation

$\lvert\mathcal{S}\rvert$	Cardinality of the set $\mathcal{S}$
$\mathbb{1}_{\mathcal{S}}$	Indicator variable over the set $\mathcal{S}$
y	Scalar y
$\lvert y\rvert$	Absolute value of y
$\boldsymbol{v}$	Vector $\boldsymbol{v}$
$\mathbf{M}$	Matrix $\mathbf{M}$
M_{ij}	The element of the matrix $\mathbf{M}$ at row i and column j
$\mathbf{M}^T$	Transpose of the matrix $\mathbf{M}$
$\lvert\mathbf{M}\rvert$	Determinant of the matrix $\mathbf{M}$
$Tr(\mathbf{M})$	Trace of the matrix $\mathbf{M}$
$\mathbb{E}(X)$	Expected value of X
$Var(X)$	Variance of X
$\log(x)$	*natural logarithm* of x (logarithm to the base e)

Chapter 1

Introduction

Networks have been studied extensively to model many interesting complex systems, including the Internet, social networks, financial networks and biological networks [10,31,34]. Any network consists of *nodes* which represent items of interest, and *edges* which represent the connectivity between pairs of nodes. For example, considering social networks, nodes are the users and the edges correspond to interactions between the users. An interesting feature many networks exhibit is *community structure*, which involves the natural dividing of nodes into groups, called *communities*, where there are denser connections within a group, and sparser connections between different groups [10, 11, 14, 31]. This particular type of community structure is also known as *assortative* [31]. For instance, social networks contain communities corresponding to real-life communities consisting of the members, such as friendship or family circles. The problem of detecting communities within networks is known as *community detection*, and algorithms are developed as a solution.

In order to provide a theoretical setting to test and compare different community detection algorithms, generative models of random graphs are very useful, and one such commonly used model is the *stochastic block model* [25, 31]. We will investigate several community detection algorithms, and will use different generative models in order to analyse and reason about them.

The underlying ingredients of the community detection algorithms have other interesting applications also, including the analysis of time series data within the context of financial networks. The aim involves seeking groups of correlated financial assets that can be used in *mean variance portfolio optimisation*. We consider the application of community

detection algorithms to real-world financial networks, in order find groups of correlated stocks found on the FTSE 100 exchange.

This project aims are twofold; firstly, we investigate and study different community detection algorithms, and secondly, apply modifications to these techniques in order to detect communities within the financial networks setting.

Chapter 2

Background

In this chapter we will describe all the technical background required to understand and detail the different settings we investigate. Initially, we will highlight some basic results in graph theory. Then, we will outline the problem of community detection and present three models used to generate random graphs with community structure to be used as a testing playground for algorithms. Following this, we will discuss basic concepts within finance required to understand the behaviour of financial assets that will provide the motivation for applying community detection algorithms to financial networks.

2.1 Graph Theory Preliminaries

We assume the reader is familiar with some basic concepts in linear algebra such as matrix multiplication, eigenvectors and eigenvalues of matrices. Rather, we will cover some basic tools within spectral graph theory using definitions from [10,12,14,24]. Spectral graph theory is the study of graphs through the eigenvalues and eigenvectors of matrices associated with the graphs [24]. We begin by defining some basic notions about graphs.

Definition 2.1. A *graph* $\mathcal{G}$ is a pair of sets (V,E), where V is a set of vertices or nodes and $E \subset V^2$, the set of unordered pairs of elements of V. The elements of E are called edges or links.

Definition 2.2. A graph $\mathcal{G} = (V, E)$ is called *undirected* if for all $v, w \in V$: $(v, w) \in E \iff (w, v) \in E$. Otherwise, $\mathcal{G}$ is called *directed*.

Definition 2.3. A *weighted* graph is a graph where a number (weight) is assigned to each edge.

We will assume, without loss of generality, that $V = \{1, \ldots, n\}$. See figure 2.1a for an example of an undirected graph with seven vertices and eleven edges.

Definition 2.4. A graph $\mathcal{G}' = (V', E')$ is a *subgraph* of $\mathcal{G} = (V, E)$ if $V' \subset V$ and $E' \subset E$. If $\mathcal{G}'$ contains all edges of $\mathcal{G}$ that join vertices of V', one says that the subgraph $\mathcal{G}'$ is induced or spanned by V'.

Definition 2.5. A partition of the vertex set V in two subsets S and $V - S$ is called a *cut*. The cut size is the number of edges of $\mathcal{G}$ joining vertices of S with vertices of $V - S$.

Definition 2.6. Two vertices are *adjacent* or *neighbours* if they are connected by an edge. The set of neighbours of a vertex v is called *neighbourhood*, and denoted by $\Gamma(v)$.

Definition 2.7. The *degree* d_v of a vertex v is the number of its neighbours, $|\Gamma(v)|$.

We will be interested in using certain graphs is the models, such as bipartite graphs.

Definition 2.8. A *bipartite* graph, is a graph whose vertices can be decomposed into two disjoint sets such that no two vertices within the same set are adjacent.

Definition 2.9. A *clique* of an undirected graph is a subset of its vertices such that every pair of vertices in the subset are adjacent in the graph.

An example of an undirected bipartite graph with nine vertices and eight edges is shown in figure 2.1b, whilst an example of a clique within an undirected graph is shown in figure 2.1c.

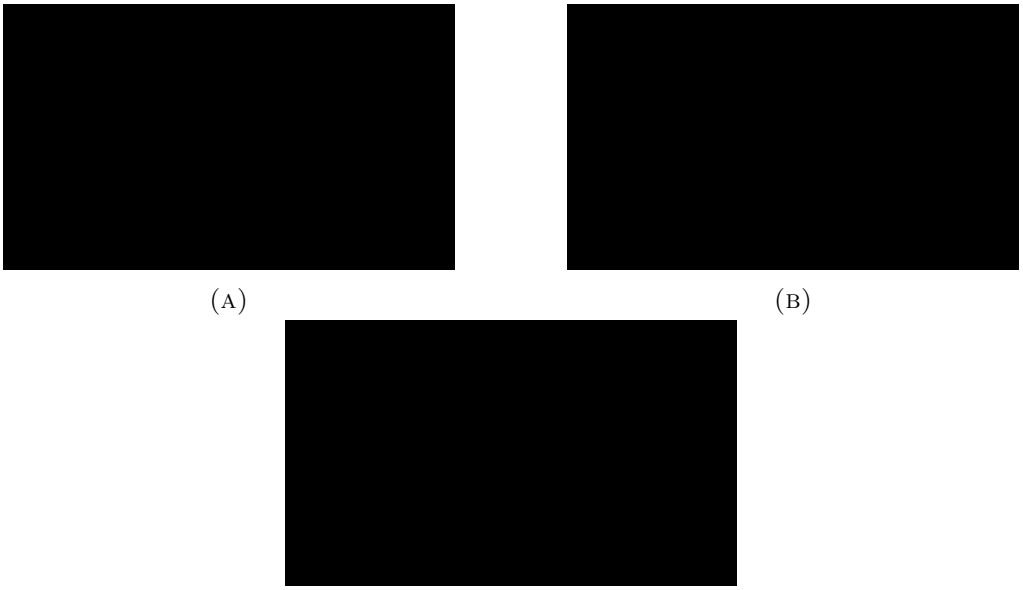
There is a very close connection between graphs and matrices, since the whole information about the topology of a graph can be entailed in matrix form, called the *adjacency matrix*.

Definition 2.10. The *adjacency matrix*, $\mathbf{A} \in \{0, 1\}^{n \times n}$ of a graph $\mathcal{G} = (V, E)$, is a $n \times n$ matrix whose element A_{ij} equals 1 if there exists an edge joining vertices i and j in $\mathcal{G}$, and zero otherwise.

From definition 2.10, it follows that elements of the adjacency matrix, $\mathbf{A}$, can be written as

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Note that the sum of elements of the i-th row of the adjacency matrix yields the degree of node i of the graph, $d_i = \sum_j A_{ij}$. Also, the adjacency matrix is symmetric if the graph is undirected.



(A) (B)

(C)

FIGURE 2.1: A set of visualisations of undirected graphs. In (a), the graph has seven nodes and eleven edges. In (b), a bipartite graph, with nine nodes (elements of disjoint sets are coloured black and red denoting membership) and eight edges, is shown. In (c), an undirected graph, with six nodes and six edges is shown, where the nodes coloured red form a clique within the graph.

Definition 2.11. The *weighted adjacency matrix*, $\mathbf{A} \in \mathbb{R}^{n \times n}$ of a weighted graph $\mathcal{G} = (V, E)$, is a $n \times n$ matrix whose element A_{ij} equals the weight of the edge connecting nodes i and j, if it exists, and zero otherwise.

There are other matrices that have also been studied extensively in spectral graph theory, including the Laplacian which is applied in topics such as graph partitioning, synchronisation and graph connectivity [14].

Definition 2.12. The *degree matrix*, $\mathbf{D}$, of a graph $\mathcal{G} = (V, E)$, is a $n \times n$ diagonal matrix whose element D_{ii} equals the degree of vertex i.

From definition 2.12, it follows that elements of the degree matrix, $\mathbf{D}$, can be written as

$$D_{ij} = \begin{cases} d_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Definition 2.13. The matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is called the *unnormalised Laplacian matrix*.

From definition 2.13, it follows that elements of the unnormalised Laplacian matrix of a graph $\mathcal{G} = (V, E)$, $\mathbf{L}$, can be written as

$$L_{ij} = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Definition 2.14. The matrix $\widetilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is called the *normalised Laplacian matrix*, where $\mathbf{I}$ is the $n \times n$ identity matrix.

Note that from definitions 2.13 and 2.14, the normalised Laplacian matrix can also be written as $\widetilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$.

An important property of adjacency and Laplacian matrices is their spectra, which we will use, later in the report, to motivate and develop a spectral clustering algorithm for community detection.

Definition 2.15. The *spectrum* of a graph $\mathcal{G}$ is the set of eigenvalues of its adjacency matrix, $\{\lambda_1, \ldots, \lambda_n\}$.

Definition 2.16. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$. The *spectral radius* is defined as $\rho(\mathbf{M}) = \max_i |\lambda_i|$.

2.2 Community Structure in Networks

An intuitive notion of communities within graphs involves the assignment of nodes to communities such that there are denser connections between nodes belonging to the same community, and sparser connections between nodes belonging to different communities. If a graph exhibits this property, it is said to contain assortative community structure [10, 11, 14, 31]. For instance, within social networks where nodes are users and edges between nodes represent interactions between the users, community structure within the graph corresponds to real-life communities consisting of the users. The aim of community detection algorithms is to estimate or recover the node assignments. The algorithms need to be efficient due to the large size of graphs in real-world networks, so we require the computationally complexity to not be worse than nearly linear in the number of edges in the graph (approximately $O(n^2 \log n)$ where n represents the number of nodes in the graph).

Within the literature, the terms *groups* and *clusters* are synonymous with communities, and as such we will use all three terms interchangeably through the report; so the reader should note all these terms refer to the same notion of communities in graphs.

In order to help provide a setting where different algorithms may be compared, we wish to study particular models which generate random graphs. One popular model is called the stochastic block model. Many special cases of this model have been studied, but we consider two versions, available in the literature. Firstly, there is a model considered by Decellle et al. [31] and Nadakuditi et al. [25], also known as the *planted partition model*. Secondly, there is a model used by Montanari [35,36], which we will refer to as the *hidden clique model*. We emphasise that we do not exclusively focus on detecting cliques for the latter model, but the name is simply convenient for reference in this report.

Let us define the stochastic block model following Decellle et al. [31]. The stochastic block model has parameters: k, n_a, $\mathbf{P}$. k represents the number of communities (or groups), n_a refers to the expected fraction of nodes within each group a, for $1 \leq a \leq k$, and a $k \times k$ matrix $\mathbf{P}$ whose element P_{ab} equals the probability of an edge occurring between nodes belonging to groups a and b. It is known as the *affinity matrix*. We proceed to generate a random directed graph, $\mathcal{G}$, consisting of n nodes. Firstly, though, assign, to each node i of the graph, $\sigma_i \in \{1, \ldots, k\}$, a label indicating which community the node belongs to. These labels are chose independently, where, for each node i, $\Pr(\sigma_i = a) = n_a$. Let $\boldsymbol{u} = [\sigma_1, \ldots, \sigma_n]^T$ be the *ground-truth node assignments* of the graph. The random graph is generated to have an adjacency matrix, $\mathbf{A}$, whose elements are defined by

$$A_{ij} = \begin{cases} 0 & \text{if } i = j \\ X & \text{otherwise} \end{cases} \tag{2.1}$$

where $X \sim Be(P_{\sigma_i, \sigma_j})$.

This formulation matches the intuition of community structure, that the connectivity between two nodes depends solely on the community memberships of the two nodes. Note, also, that we do not allow self-loops.

The framework for testing community detection algorithms, which we will use, can now be summarised. Firstly, we generate synthetic datasets, by creating random graphs from the models described in sections 2.2.1 and 2.2.2, with varying parameters and known underlying ground-truth node assignments. Then we use the synthetically-generated graphs as an input to various algorithms (an appropriate model is chosen for each algorithm), which provides, as output, an estimate to the community assignments. Finally, for each

algorithm, we compare the estimated community assignments to the ground-truth values. This provides a notion of performance and accuracy to compare between the algorithms.

We now describe two models; one is a special case of the stochastic block model, created by imposing specific properties on the parameters, and the other a slightly modified version.

2.2.1 Planted Partition Model

We will consider the formulation of the planted partition model as given by Decellle et al. [31]. To construct the planted partition model, we consider the stochastic block model, with $n_a = 1/k$, and the affinity matrix defined by

$$P_{ab} = \begin{cases} p_{in} & \text{if } a = b \\ p_{out} & \text{otherwise} \end{cases} \tag{2.2}$$

These properties essentially result approximately equal number of nodes and edges within each community. From this definition, p_{in} represents the probability of an edge occurring between two nodes belonging to the same community and p_{out} represents the probability of an edge occurring between two nodes belonging to the different communities. We assume assortative structure so that $p_{in} > p_{out}$. This just matches the intuition of edges more likely to appear between nodes belonging to the same community than between nodes belonging to different communities.

An example of a random graph generated by the planted partition model is shown in figure 2.2a. We labelled nodes using $\sigma_i = 1 + (i \bmod k)$ for $i = 1, \ldots, n$ and generated the graph with $n = 300$, $k = 3$, $p_{in} = 0.7$, $p_{out} = 0.3$. The adjacency matrix of this graph is plotted with a pixel shaded red if the element in the adjacency matrix, corresponding to the location of the pixel, equals 1; while a pixel is shaded white if the element equals 0. Since we know the ground truth labelling of nodes, we can, without loss of generality, reorder the rows and columns of the adjacency matrix, such that it consists of blocks of nodes associated with the node community memberships. This is plotted in figure 2.2b. Note that since $k = 3$, there are $3 \times 3 = 9$ blocks, where the blocks are denser along the main diagonal since these correspond to edges between nodes belonging to the same community and $p_{in} > p_{out}$.

In figure 2.3, we show a visualisation of an instance of a random graph generated by the planted partition model with parameters with $n = 240$, $k = 3$, $p_{in} = 0.2$, $p_{out} = 0.01$.

(A) (B)

FIGURE 2.2: A set of plots for unlabelled, (a), and labelled, (b), adjacency matrices for random graph generated by planted partition model. The graphs were generated with $n = 300$, $k = 3$, $p_{in} = 0.7$ and $p_{out} = 0.3$.

FIGURE 2.3: A visualisation of an instance of a random graph generated by the planted partition model with $n = 240$, $k = 3$, $p_{in} = 0.2$ and $p_{out} = 0.01$.

Decelle et al. [19] conjectured a phase transition for sparse graphs generated from the planted partition model, using non-rigorous ideas from statistical physics [27]. Nadakuditi et al. [25] used methods from random matrix theory to present an asymptotic analysis of spectra of random graphs to also demonstrate the presence of a phase transition. Essentially, we can distinguish between a *detectable* phase where it is possible to learn node assignments in a way that is correlated with the ground-truth node assignments of the graph, and an *undetectable* phase, where learning is impossible.

Let us define, for convenience, the variables $c_{in} = np_{in}$ and $c_{out} = np_{out}$. Consider a graph, generated by the planted partition model, and following the argument of [25], which we will not explain, one finds a transition occurring at the point

$$c_{in} - c_{out} = \sqrt{k(c_{in} + (k-1)c_{out})}. \tag{2.3}$$

In particular, let us consider the case where $k = 2$, so we find a transition at

$$c_{in} - c_{out} = \sqrt{2(c_{in} + c_{out})}. \tag{2.4}$$

Mossel, Neeman and Sly [27] proved the undetectable phase region of the conjecture given

by equation equation (2.4) (that is to say, it is impossible to meaningfully recover the node assignments when $c_{in} - c_{out} < \sqrt{2(c_{in} + c_{out})}$. Massoulié [32] and then, independently using a different proof, Mossel et al. [33] proved the detectable phase region of the conjecture, meaning it is possible to recover node assignments positively correlated with the ground-truth when $c_{in} - c_{out} > \sqrt{2(c_{in} + c_{out})}$. The techniques used to prove these results are beyond the scope of this report, however these results provide a very important limit on the ability of algorithms to detect communities. This motivates the development of algorithms which can efficiently (in nearly linear time) detect communities, in the sparse regime, up to the limit.

2.2.2 Hidden Clique Model

We will consider the following formulation as explained by Montanari [35,36]. To construct the hidden clique model, we consider the stochastic block model with some modifications. We proceed to generate a graph with n nodes and k communities but with the affinity matrix, $\mathbf{P}$, becoming a $(k+1) \times (k+1)$ matrix defined by

$$P_{ab} = \begin{cases} p_{in} & \text{if } a = b,\ a \leq k,\ b \leq k \\ p_{out} & \text{otherwise} \end{cases} \tag{2.5}$$

Another tweak is that we now consider the variable n_a to represent the number of nodes within community a (rather than the expected fraction of nodes). Note that the number of nodes within each community does not necessarily sum to n, since we consider them to be 'hidden' within the graph. Also, we are interested in the regime where the size of these communities is small relative to the size of the graph. Once more, we assume assortative structure within the hidden communities so that $p_{in} > p_{out}$.

An example of a random graph generated by the hidden clique model is shown in figure 2.4a. We generated the graph with $n = 300$, $k = 3$, $p_{in} = 0.8$, $p_{out} = 0.2$, $n_1 = 50$, $n_2 = 40$, $n_3 = 20$. The adjacency matrix of this graph is plotted with a pixel shaded red if the element in the adjacency matrix, corresponding to the location of the pixel, equals 1; while a pixel is shaded white if the element equals 0. Since we know the ground truth labelling of nodes, we can, without loss of generality, reorder the rows and columns of the adjacency matrix, such that it consists of blocks of nodes associated with the node community memberships. This is plotted in figure 2.4b. We can see three dense blocks of size 50, 40 and 20 respectively, corresponding to the three hidden communities.

(A) (B)

FIGURE 2.4: A set of plots for unlabelled, (a), and labelled, (b), adjacency matrices for random graph generated by hidden clique model. The graphs were generated with $n = 300$, $k = 3$, $p_{in} = 0.8$, $p_{out} = 0.2$, $n_1 = 50$, $n_2 = 40$, $n_3 = 20$.

FIGURE 2.5: A visualisation of an instance of a random graph generated by the hidden clique model with $n = 150$, $k = 1$, $n_1 = 30$, $p_{in} = 1.0$, $p_{out} = 0.1$. Nodes belonging to the hidden community, which in this case is in fact a clique, are coloured red whilst other nodes are coloured blue.

In figure 2.5, we show a visualisation of an instance of a random graph generated by the hidden clique model with parameters with $n = 150$, $k = 1$, $n_1 = 30$, $p_{in} = 0.3$, $p_{out} = 0.05$.

An interesting phase transition can also be derived for these models also. We follow the argument of Montanari [35, 36] to show this. We consider the simplest case of the model with only one hidden community. We begin by generating a graph from the hidden clique model with n nodes and one community (i.e. $k = 1$). Assume the n_1 nodes belonging to the hidden community make up a hidden community set, $\mathcal{S}$ (i.e. $|\mathcal{S}| = n_1$). Define $\mathbb{1}_n \in \{1\}^n$ as the n-dimensional vector with every element equal to 1. Also let $\mathbb{1}_{\mathcal{S}}$ be the indicator variable for nodes belonging to the hidden community. Denote the adjacency matrix of the graph by $\mathbf{A}$, where each element is defined by

$$A_{ij} \sim Be(p_{ij}) \tag{2.6}$$

where

$$p_{ij} = \begin{cases} p_{in} & \text{if } i \in \mathcal{S},\ j \in \mathcal{S} \\ p_{out} & \text{otherwise} \end{cases} \tag{2.7}$$

Then, simply, we get the following

$$\mathbb{E}(\mathbf{A}) = (p_{in} - p_{out})\mathbb{1}_{\mathcal{S}}\mathbb{1}_{\mathcal{S}}^T + p_{out}\mathbb{1}_n\mathbb{1}_n^T \tag{2.8}$$

and

$$Var(A_{ij}) = p_{out}(1 - p_{out}) \text{ if } \{i, j\} \not\subseteq \mathcal{S} \tag{2.9}$$

Denote $\widetilde{\mathbf{A}}$ as the *normalised adjacency matrix* of the graph, defined by

$$\widetilde{\mathbf{A}} \equiv \frac{1}{\sqrt{np_{out}(1 - p_{out})}}(\mathbf{A} - p_{out}\mathbb{1}_n\mathbb{1}_n^T) \tag{2.10}$$

By taking the expectation and using equation (2.8), we get the following

$$\mathbb{E}(\widetilde{\mathbf{A}}) = \frac{1}{\sqrt{np_{out}(1 - p_{out})}}(p_{in} - p_{out})\mathbb{1}_{\mathcal{S}}\mathbb{1}_{\mathcal{S}}^T \tag{2.11}$$

Let us write $\widetilde{\mathbf{A}} = \mathbb{E}(\widetilde{\mathbf{A}}) + (\widetilde{\mathbf{A}} - \mathbb{E}(\widetilde{\mathbf{A}}))$. Now using equation (2.11), we get

$$\widetilde{\mathbf{A}} = \frac{1}{\sqrt{np_{out}(1 - p_{out})}}(p_{in} - p_{out})\mathbb{1}_{\mathcal{S}}\mathbb{1}_{\mathcal{S}}^T + (\widetilde{\mathbf{A}} - \mathbb{E}(\widetilde{\mathbf{A}})) \tag{2.12}$$

Let us now define the following

$$\lambda \equiv \frac{p_{in} - p_{out}}{\sqrt{np_{out}(1 - p_{out})}} \tag{2.13}$$

$$\boldsymbol{u} \equiv \mathbb{1}_{\mathcal{S}} \tag{2.14}$$

$$\mathbf{Z} \equiv \widetilde{\mathbf{A}} - \mathbb{E}(\widetilde{\mathbf{A}}) \tag{2.15}$$

Then we can re-write equation (2.12) as

$$\widetilde{\mathbf{A}} = \lambda \boldsymbol{u}\boldsymbol{u}^T + \mathbf{Z} \tag{2.16}$$

One can interpret λ as a signal-to-noise ratio, $\boldsymbol{u}$ as a signal (i.e. the ground-truth node assignments we which to infer) and $\mathbf{Z}$ as zero-mean noise with i.i.d. entries. We have essentially represented the problem of inferring the hidden community from the graph by a problem of estimating a rank-1 matrix in noise. Notice that for the generalisation with k communities, we would get a rank-k matrix plus noise for the normalised adjacency matrix.

2.3 Financial Networks

2.3.1 Prices and Returns of Financial Assets

Financial assets are instruments claiming to have monetary value that can be bought and sold. Financial assets can be separated into broad classes, with examples including stocks, bonds or real estate [22, 37]. The values of these assets is reflected in their price, which varies with time. Investors may wish to decide between which asset classes to invest in at any time, a process known as *asset allocation* [37]. Also, within a particular asset class, an investor wishes to allocate money to specific assets, a process known as *portfolio selection* [37]. For this report, we will focus on portfolio selection of stocks in our application of community detection algorithms, due to available data constraints; however a very similar scheme may be used to tackle asset allocation also.

Investors tend not to consider the prices of assets they have invested in, but rather the *return* generated. Let us consider a financial asset whose price at time t is $p(t)$. One popular measure of return is called the *rate of return* [23, 37] at a time t, denoted by $r(t)$, which is defined as

$$r(t_0) = \frac{p(t_1) - p(t_0)}{p(t_0)} \tag{2.17}$$

where we can interpret t_1 as the time when the investor sold the asset, and t_0 as the time when the investor bought the asset.

Critically, the rate of return is sensitive to large changes for longer time horizons [5]. In particular, we can consider a different measure of return, which is equivalent to a return with a constant interest rate [5]. We can generalise the concept interest rates with the simple example of an investor placing money (investing) in a bank account, as explained in [2, 21]. The amount of money initially invested is referred to as the *principal*. We then assume money grows by a multiplicative factor, where the gain is paid into the account by the bank. This process is often called *compounding*. The time at which the interest is compounded, is called the compounding period. *Compound interest* involves interest being paid on both the principal and the accumulated interest up to the present [5]. Typically, we are interested in the number of compounding periods in one year (i.e. the number of times the interest on our principal is compounded each year) [21]. Denote the principal by w_0, the amount in the account time t by w_t, the interest rate by y and the number of compounding periods in a year by m. Then the amount within the account

holdings after 1 year is given by

$$w_1 = w_0(1 + (y/m))^m \tag{2.18}$$

We can imagine diving a year into infinitesimally small compounding periods, and then determine the effect of this continuous compounding by taking the limit of ordinary compounding [2]. Notice the total number of compounding periods in a length of t years is given by mt. Thus the effect of continuous compounding is

$$w_t = \lim_{m\to\infty} w_0(1 + (y/m))^{mt} = w_0 \exp(yt) \tag{2.19}$$

If we divide equation (2.19) by the initial investment w_0, and take the natural logarithm, we get a representation for the return, $r = yt$. This indicates that taking the natural logarithm results in a constant interest rate. We have arrived at another measure for return, called the *logarithmic return*, that is defined as

$$r(t_0) = \log(p(t_1)) - \log(p(t_0)) \tag{2.20}$$

where, once more, we can interpret t_1 as the time when the investor sold the asset, and t_0 as the time when the investor bought the asset.

There are several advantages to using logarithmic returns, as explained in [40], which we will briefly summarise. Firstly, if we assume asset prices have a *log normal* distribution, then the logarithmic returns are conveniently normally distributed [40]. The reasons why assuming a log normal distribution may be appropriate for dynamic pricing of assets is beyond the scope of this report. Secondly, for small rates of return, the logarithmic return is approximately equal to the rate of return [40]. To see this, consider the following approximation combined with equation (2.17) and equation (2.20)

$$\log(1 + x) \approx x, \text{ for } x \ll 1 \tag{2.21}$$

Thirdly, we benefit from numerical stability since the addition of small numbers is numerically stable, whilst multiplying small numbers is subject to *arithmetic underflow* [40].

However, there are disadvantages to using the logarithmic return, including the issue that the derivation is only correct if the interest rate is constant [5, 40]. Nonetheless, the logarithmic return is widely used in the literature (e.g. see [5–7, 17, 20, 34]), and hence we shall use it for the rest of the report, and the reader should note we shall use the terms

FIGURE 2.6: Plots of prices and logarithmic returns for Anglo American plc (AAL) between 2004 and 2014

'return' and 'logarithmic return' interchangeably from now on. An example plot of price and logarithmic return for a stock is shown in figure 2.6.

2.3.2 Mean-Variance Portfolio Theory

The term *portfolio* relates to a investing in a combination of different assets. We can characterise a portfolio by *portfolio weights*, where the weight of an asset within the portfolio is given by the ratio of the value of the position in the asset divided by the total value of the portfolio. We are particularly interested in the return of the portfolio, which is related to the mean of the returns of the individual assets that make up the portfolio, as we will see, and the risk of the portfolio, which is related to the variance of the returns of the individual assets. This is the source of the term mean-variance portfolio theory. The reader should realise the inherit trade-off between risk and return; if the investor wishes to realise a larger return, he must bear a higher risk. A more detailed explanation of this relationship is given in [2, 23]. Rather we are simply interested in finding the most efficient, or *minimum-variance portfolio* for a given requested portfolio return (that is to say, the investor requests a specific expected return, and wishes to form a portfolio that has the lowest variance of all possible portfolio that can deliver the specified expected return). We can formalise the mean-variance portfolio setting in the following way, which summarises the explanations from [2, 5, 23].

Let P be a portfolio comprising of n assets, where the return of the portfolio is denoted by r_P and the variance of the portfolio is denoted by σ_P^2. Denote the return of asset i by r_i, the variance of the asset's return by $\sigma_i^2 \equiv Var(r_i)$, and the weight of asset i in the portfolio by w_i. Also let X_0 denote the total amount invested in the portfolio (i.e. initial investment in the portfolio by the investor) and X_{0i} represent the the amount invested in

asset i. We select the amounts in the assets forming the portfolio such that

$$\sum_{i=1}^{n} X_{0i} = X_0 \tag{2.22}$$

We define the portfolio weights using

$$w_i = \frac{X_{0i}}{X_0},\ i = 1, \ldots, n \tag{2.23}$$

and clearly,

$$\sum_{i=1}^{n} w_i = 1 \tag{2.24}$$

Notice that a negative weight indicates a *short position* in that asset, and that the returns of the individual assets and portfolio are random variables.

In particular we can represent the return of the portfolio by

$$r_P = \sum_{i=1}^{n} w_i r_i \tag{2.25}$$

By using equation (2.25), we obtain the expected return of the portfolio

$$\mathbb{E}(r_P) = \sum_{i=1}^{n} w_i \mathbb{E}(r_i) \tag{2.26}$$

and the variance of the portfolio return

$$Var(r_P) \equiv \sigma_P^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \rho_{ij} \sigma_i \sigma_j \tag{2.27}$$

where ρ_{ij} is the correlation coefficient of the returns of assets i and j, defined as

$$\rho_{ij} = \frac{Cov(r_i, r_j)}{\sigma_i \sigma_j} \tag{2.28}$$

Note that the standard deviation of the returns of the portfolio (or any asset) is often called its *volatility*.

This formulation serves a key question, given estimates of each assets returns, variances and covariances (which one can obtain from historical data), how does one pick a selection of these assets, for any given time period, in order to form the best portfolio for the investor? We see in equation (2.27), that by simply investing in assets which have a lower

correlation with one another (i.e. a lower value of ρ_{ij}), we can reduce the variance, and thus risk, of the portfolio. This process is called *diversification*. Thus, for any given time period (and possibly dynamically), finding groups of assets, where the returns have higher correlation within groups and lower correlation between groups would help by presenting 'baskets' of assets that the investor can pick from knowing selecting from a range of baskets would be beneficial (of course which assets to select from inside the basket relates to the risk-return trade off). This serves as the main motivation for the application of community detection algorithms within financial networks, as we formalise in section 2.3.3.

2.3.3 Constructing Financial Networks

From section 2.3.2 we understand one way to help minimise risk in constructing portfolios involves analysing the correlation coefficients of returns between two assets. In order to study all possible correlations between all available assets, we construct a weighted, undirected and fully-connected network of assets, which we call the *financial network*. The following model has been considered by [4, 6, 7, 17, 34].

Let us consider the situation where the investor is faced with n financial assets, and has access to historical price data for all these assets for T time steps. The time steps may be trading days, or weeks, for instance, and the appropriate choice will depend on the type of assets available.

We proceed to construct a graph with n nodes, where each nodes represents an asset, and assign to the i-th node a single time series, denoted by X_i, which is defined as

$$X_i = \{x_i(1), \ldots, x_i(T)\} \tag{2.29}$$

where $x_i(t)$ describes the logarithmic return of asset i at time t, defined by equation (2.20). This time series describes the evolution of the logarithmic return of the asset over T time steps. We then model the weight of an edge connecting nodes i and j of the graph by the cross-correlation between the time series corresponding to assets i and j. We form a cross-correlation matrix, denoted by $\mathbf{C}$, whose elements are defined by

$$C_{ij} = \frac{\langle X_i X_j \rangle - \langle X_i \rangle \langle X_j \rangle}{\sqrt{[\langle X_i^2 \rangle - \langle X_i \rangle^2][\langle X_j^2 \rangle - \langle X_j \rangle^2]}} \tag{2.30}$$

FIGURE 2.7: Example of a correlation matrix. Evaluated from an ensemble of 80 stocks listed on the FTSE 100 (see appendix A for a list) using data between 01/01/2011 and 01/01/2013.

where the $\langle \cdots \rangle$ notation denotes a time average, so that

$$\langle X_i \rangle = \frac{1}{T} \sum_{t=1}^{T} x_i(t) \tag{2.31}$$

$$\langle X_i^2 \rangle = \frac{1}{T} \sum_{t=1}^{T} x_i^2(t) \tag{2.32}$$

$$\langle X_i X_j \rangle = \frac{1}{T} \sum_{t=1}^{T} x_i(t) x_j(t) \tag{2.33}$$

We also assume each time series X_i has been standardised (before assigning to the node) by using

$$X_i \coloneqq \frac{X_i - \langle X_i \rangle}{\sqrt{\langle X_i^2 \rangle - \langle X_i \rangle^2}} \tag{2.34}$$

so that

$$\langle X_i \rangle = 0 \tag{2.35}$$

$$\langle X_i^2 \rangle - \langle X_i \rangle^2 = 1 \tag{2.36}$$

Note that the cross correlation values is just a sample estimate for the correlation coefficient, ρ_{ij}, used in section 2.3.3, calculated from the historical data.

We can then characterise the financial network by the correlation matrix, which we also refer to as the networks weighted adjacency matrix.

In figure 2.7, we have plotted a correlation matrix using data of 80 stocks listed on the FTSE 100 (see appendix A for a list) between 2011 and 2013. Notice that the main diagonal has all elements equal to one, as you would expect, and that there are very few negative elements (i.e. very few assets that are anti-correlated with one another).

The problem statement can now summarised. Given a financial network, how can we group nodes into communities where correlations are higher within the communities and lower between the communities? Contrary to graphs with community structure described in section 2.2, the weights of the edges rather than the topology of the network are crucial in determining community memberships. In other words, we focus solely on the weighted adjacency matrix of the graph. Also the reader should note that the correlations between asset returns will vary over time, and thus representing this relationship dynamically (rather than over a one long period of time) is very important since investors wish to change their positions in order to react to the dynamics of market conditions.

Chapter 3

Community Detection in Financial Networks

In this chapter we aim to apply algorithms to detect communities within real-world financial networks. Firstly, we explain the process of gathering the data to create the financial networks. Then we investigate and apply different community detection algorithms and compare their performance on both synthetically generated and the real-world data.

3.1 Constructing the Real-world Financial Network

The dataset we use consists of daily closing prices of 80 stocks in the FTSE 100 index, which we obtained from [39]. The time period considered is between the beginning of 2004 to the end of 2013, a total of 2501 prices. his is the data we obtained after the removal of a few data points due to incomplete data across different stocks.The complete list of stocks is given in appendix A. We then calculated, for each stock and for each time period, the logarithmic return. We generated a time series of these returns and associated each stock with a single time series. By using the method described in section 2.3.3, we proceeded to construct financial network represented by a fully-connected, undirected and weighted graph. There are 80 nodes in this network (each one representing one of the stocks), and the weights on the edges connecting any two nodes is the cross-correlation between the time series of returns associated with the stocks represented by the two nodes. Let us denote the cross-correlation matrix of this network, defined in equation (2.30), by $\mathbf{C}$. We

stress, at this point, the data of the whole period (01/01/2004 - 01/01/2013) is currently represented by this single network.

3.1.1 Random Matrix Theory

3.2 Community Detection Algorithms

3.2.1 Modularity Optimisation Methods

3.2.2 Spectral Clustering Algorithm

3.2.3 Modified Modularity Optimisation Method

3.3 Synthetic Data Testing

3.4 Application to Real-world Financial Network

Appendix A

List of Stocks

TABLE A.1: The ticker symbol, name and industry of the 80 FTSE 100 companies studied. Classifications were obtained from [38, 39].

Ticker	Name	Industry
AAL	Anglo American	Basic Materials
ABF	Associated British Foods	Consumer Goods
ADN	Aberdeen Asset Management	Financials
AGK	Aggreko	Industrials
AHT	Ashtead Group	Industrials
AMEC	AMEC	Basic Materials
ANTO	Antofagasta	Basic Materials
ARM	ARM Holdings	Technology
AV	Aviva	Financials
AZN	AstraZeneca	Health Care
BAB	Babcock International Group	Industrials
BARC	Barclays	Financials
BATS	British American Tobacco	Consumer Goods
BG	BG Group	Basic Materials
BLND	British Land Company	Technology
BLT	BHP Billiton	Basic Materials
BNZL	Bunzl	Industrials
BP	BP	Basic Materials
BRBY	Burberry Group	Consumer Goods
BSY	British Sky Broadcasting Group	Consumer Services
BT-A	BT Group	Technology

CCL	Carnival	Consumer Goods
CNA	Centrica	Utilities
CPI	Capita	Industrials
DGE	Diageo	Consumer Goods
EZJ	easyJet	Consumer Services
GFS	G4S	Industrials
GKN	GKN	Consumer Goods
GSK	GlaxoSmithKline	Health Care
HMSO	Hammerson	Financials
HSBA	HSBC Holdings	Financials
IAG	International Consolidated Airlines Grp	Consumer Services
IHG	InterContinental Hotels Group	Consumer Services
IMT	Imperial Tobacco Group	Consumer Goods
ITRK	Intertek Group	Industrials
ITV	ITV	Consumer Services
JMAT	Johnson Matthey	Basic Materials
KGF	Kingfisher	Consumer Services
LAND	Land Securities Group	Financials
LGEN	Legal & General Group	Financials
LLOY	Lloyds Banking Group	Financials
LSE	London Stock Exchange Group	Financials
MGGT	Meggitt	Industrials
MKS	Marks and Spencer Group	Consumer Services
MRO	Marathon Oil Corporation	Oil & Gas
MRW	Wm. Morrison Supermarkets	Consumer Services
NG	National Grid	Utilities
NXT	NEXT	Consumer Services
OML	Old Mutual	Financials
PRU	Prudential	Financials
PSN	Persimmon	Industrials
PSON	Pearson	Consumer Services
RB	Reckitt Benckiser Group	Consumer Goods
RBS	Royal Bank of Scotland Group	Financials
RDSB	Royal Dutch Shell	Basic Materials
REL	Reed Elsevier	Consumer Services
REX	Rexam	Consumer Goods

RIO	Rio Tinto	Basic Materials
RR	Rolls-Royce Holding	Industrials
RRS	Randgold Resources Limited	Basic Materials
SBRY	J Sainsbury	Consumer Services
SDR	Schroders	Financials
SGE	The Sage Group	Technology
SHP	Shire	Health Care
SMIN	Smiths Group	Industrials
SN	Smith & Nephew	Health Care
SSE	SSE	Utilities
STAN	Standard Chartered	Financials
SVT	Severn Trent	Utilities
TATE	Tate & Lyle	Consumer Goods
TPK	Travis Perkins	Industrials
TSCO	Tesco	Consumer Services
ULVR	Unilever	Consumer Goods
UU	United Utilities Group	Utilities
VOD	Vodafone Group	Technology
WEIR	The Weir Group	Industrials
WMH	William Hill	Consumer Services
WOS	Wolseley	Industrials
WPP	WPP ORD 10P	Consumer Services
WTB	Whitbread	Consumer Services

Bibliography

[1] E. P. Wigner, "Characteristic vectors of bordered matrices with infinite dimensions," in *The Annals of Mathematics, 62*, pp. 548–564, 1955.

[2] D. G. Luenberger, "Investment science," 1998.

[3] A. M. Sengupta and P. P. Mitra, "Distributions of singular values for some random matrices," *Phys. Rev. E*, vol. 60, pp. 3389–3392, 1999.

[4] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, and H. E. Stanley, "Universal and non-universal properties of cross-correlations in financial time series," 1999, arXiv:cond-mat/9902283v1.

[5] J.-P. Onnela, "Taxonomy of Financial Assets." `http://jponnela.com/`, 2002.

[6] J.-P. Onnela, A. Chakraborti, K. Kaski, and J. Kertesz, "Dynamic asset trees and portfolio analysis," 2002, arXiv:cond-mat/0208131v1.

[7] J.-P. Onnela, K. Kaski, and J. Kertesz, "Clustering and information in correlation based financial networks," 2003, arXiv:cond-mat/0312682v1.

[8] M. E. J. Newman, "The structure and function of complex networks," 2003, arXiv:cond-mat/0303516v1.

[9] M. Potters, J. Bouchaud, and L. Laloux, " Financial Applications of Random Matrix Theory: Old Laces and New Pieces," 2005, arXiv:physics/0507111v1.

[10] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," 2006, arXiv:physics/0605087v3.

[11] M. E. J. Newman, "Modularity and community structure in networks," 2006, arXiv:physics/0602124v1.

[12] D. Spielman, "Spectral graph theory and its applications," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pp. 29–38, 2007.

[13] L. K. Chan, J. Lakonishok, and B. Swaminathan, "Industry Classifications and Return Comovement," in *Financial Analysts Journal, Vol. 63, No. 6*, 2007.

[14] S. Fortunato, "Community detection in graphs," 2010, arXiv:0906.0612v2.

[15] J. Leskovec, K. J. Lang, and M. W. Mahoney, "Empirical Comparison of Algorithms for Network Community Detection," 2010, arXiv:1004.3539v1.

[16] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," 2010, arXiv:1008.3926v1.

[17] D. J. Fenn, M. A. Porter, P. J. Mucha, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones, "Dynamical Clustering of Exchange Rates," 2010, arXiv:0905.4912v2.

[18] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," 2011, arXiv:1001.3448v4.

[19] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborova, "Phase transition in the detection of modules in sparse networks," 2011, arXiv:1102.1182v1.

[20] D. J. Fenn, M. A. Porter, S. Williams, M. McDonald, N. F. Johnson, and N. S. Jones, "Temporal Evolution of Financial Market Correlations," 2011, arXiv:1011.3225v2.

[21] D. Kuhn, "The basic theory of interest," *Computational Finance: 422 Lecture Notes*, 2012.

[22] D. Kuhn, "Fixed-income securities," *Computational Finance: 422 Lecture Notes*, 2012.

[23] D. Kuhn, "Mean variance portfolio theory," *Computational Finance: 422 Lecture Notes*, 2012.

[24] D. Spielman, "Spectral graph theory and its applications," in *Combinatorial Scientific Computing. Chapman and Hall / CRC Press*, 2012.

[25] R. R. Nadakuditi and M. E. J. Newman, "Graph spectra and the detectability of community structure in networks," 2012, arXiv:1205.1813v1.

[26] S. Ertekin, H. Hirsh, and C. Rudin, "Learning to Predict the Wisdom of Crowds," 2012, arXiv:1204.3611v1.

[27] E. Mossel, J. Neeman, and A. Sly, "Stochastic Block Models and Reconstruction," 2012, arXiv:1202.1499v4.

[28] J. Yang and J. Leskovec, "Defining and Evaluating Network Communities based on Ground-truth," 2012, arXiv:1205.6233v3.

[29] E. Mossel, J. Neeman, and A. Sly, "Belief Propagation, Robust Reconstruction, and Optimal Recovery of Block Models," 2013, arXiv:1309.1380v1.

[30] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborova, and P. Zhang, "Spectral redemption: clustering sparse networks," 2013, arXiv:1306.5550v2.

[31] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborova, "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications," 2013, arXiv:1109.3041v2.

[32] L. Massoulie, "Community detection thresholds and the weak Ramanujan property," 2013, arXiv:1311.3085v1.

[33] E. Mossel, J. Neeman, and A. Sly, "A Proof Of The Block Model Threshold Conjecture," 2013, arXiv:1311.4115v1.

[34] M. MacMahon and D. Garlaschelli, "Unbiased community detection for correlation matrices," 2013, arXiv:1311.1924v1.

[35] Y. Deshpande and A. Montanari, "Finding Hidden Cliques of Size sqrt(N/e) in Nearly Linear Time," 2013, arXiv:1304.7047v1.

[36] "Talk at simons institute for the theory of computing: Finding small structures in large datasets." `http://simons.berkeley.edu/talks/andrea-montanari-2013-11-18`, cited in March 2014.

[37] Z. Bodie, A. Kane, and A. J. Marcus, "Essentials of investments ninth edition," 2013.

[38] "Wikipedia: Industry classifications benchmark." `http://en.wikipedia.org/wiki/Industry_Classification_Benchmark`, cited in March 2014.

[39] "Yahoo finance website." `http://uk.finance.yahoo.com/`, cited in March 2014.

[40] "Quantivity wordpress blog." `http://quantivity.wordpress.com/2011/02/21/why-log-returns/`, cited in March 2014.

[41] Y. Deshpande and A. Montanari, " Information-theoretically Optimal Sparse PCA," 2014, arXiv:1402.2238v1.