Clustering in Financial Markets

A Network Theory Approach

KRISTINA SÖRENSEN

Master of Science Thesis
Stockholm, Sweden 2014

Clustering in Financial Markets

A Network Theory Approach

KRISTINA SÖRENSEN

Master's Thesis in Optimization and Systems Theory (30 ECTS credits)
Master Programme in Mathematics (120 credits)
Royal Institute of Technology year 2014
Supervisor at University of Florida, was Panos M. Pardalos
Supervisor at KTH was Johan Karlsson
Examiner was Johan Karlsson

TRITA-MAT-E 2014:54
ISRN-KTH/MAT/E--14/54--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

In this thesis we consider graph partition of a particular kind of complex networks referred to as power law graphs. In particular, we focus our analysis on the market graph, constructed from time series of price return on the American stock market. Two different methods originating from clustering analysis in social networks and image segmentation are applied to obtain graph partitions and the results are evaluated in terms of the structure and quality of the partition. Along with the market graph, power law graphs from three different theoretical graph models are considered. This study highlights topological features common in many power law graphs as well as their differences and limitations.

Our results show that the market graph possess a clear clustered structure only for higher correlation thresholds. By studying the internal structure of the graph clusters we found that they could serve as an alternative to traditional sector classification of the market. Finally, partitions for different time series was considered to study the dynamics and stability in the partition structure. Even though the results from this part were not conclusive we think this could be an interesting topic for future research.

Keywords: *Complex networks, cluster analysis, graph partition, market graph, power law graphs, random graphs.*

Sammanfattning

I denna uppsats studeras graf partition av en typ av komplexa nätverk som kallas power law grafer. Specifikt fokuserar vi på marknadengrafen, konstruerad av tidsserier av aktiepriser på den amerikanska aktiemarknaden. Två olika metoder, initialt utvecklade för klusteranalys i sociala nätverk samt för bildanalys appliceras för att få graf-partitioner och resultaten utvärderas utifrån strukturen och kvaliten på partitionen. Utöver marknadsgrafen studeras även power law grafer från tre olika teoretiska grafmodeller. Denna studie belyser topologiska egenskaper vanligt förekommande i många power law grafer samt modellerns olikheter och begränsningar.

Våra resultat visar att marknadsgrafen endast uppvisar en tydlig klustrad struktur för högre korrelation-trösklar. Genom att studera den interna strukturen hos varje kluster fann vi att kluster kan vara ett alternativ till traditionell marknadsindelning med industriella sektorer. Slutligen studerades partitioner för olika tidsserier för att undersöka dynamiken och stabiliteten i partitionsstrukturen. Trots att resultaten från denna del inte var entydiga tror vi att detta kan vara ett intressant spår för framtida studier.

Nyckelord: *Komplexa nätverk, klusteranalys, graf partition, marknadsgrafen, power law grafer, slumpmässiga grafer.*

Acknowledgements

I would like to thank Dr. Panos M. Pardalos and the Center for Applied Optimization, University of Florida for inviting me and giving me the chance to write my thesis abroad. A special thanks to Dr. Pardalos for his guidance and the interesting discussions during the past months. Also, I want to express my gratitude to my supervisor Johan Karlsson at the Institution of Optimization and Systems theory, KTH for all his help and advice throughout the process.

Kristina Sörensen, Stockholm 04/09/14

Contents

1

Introduction

1.1 Background

Financial analysis of today often involve interpretation of very large data sets. One convenient way to represent this large amount of data is in terms of a network. Network theory has been used to analyse many different concepts, examples span from Internet and social networks to biological networks, and recently financial networks.

Despite arising from different fields many of these network share topological characteristics which cannot be described by neither uniform random graphs nor by regular lattices. Thus to describe the complex topology of these graphs a new field emerged, complex network theory. One feature observed in many of these networks is the occurrence of a heavy tail in the degree distribution. A network showing this characteristic is called a scale free network or a power law graph. Another common feature in these networks is their tendency to form clustered communities in the graph. This introduces new problems to find specific clusters or partitions of the networks into different clusters.

Several models for representing financial networks have been proposed. Results from previous research revealed overall structure of the market as well as introduced a tool for studying market dynamics [1]. Other considered topics involve the grouping of instruments, stock classification and finding highly influential actors in the market [2]. Many previous studies have also focused on identifying specific substructures in the graph [3]. One such example is the maximum clique problem, i.e. to identify a complete sub graph of maximal cardinality in the graph. However, as many other network optimization problem this is NP-hard which often makes it impossible to find an exact solution in a reasonable amount of time.

1.2 Statement of purpose

The aim of this paper is to study community partition of the market graph. The partitions will be obtained by using two different, well known objective functions for graph partition. The resulting optimization problems will be presented together with heuristic approaches to solve two partition formulations. Additional to the empirical market graph, graph partitions of genetic power law graphs will be studied. The motivation for this is to compare the partition structure of the market graph with some theoretical models for power law random graphs. Each power law graph model will be presented and followed by an empirical study of the topological structure of some graph instances. Consequently, the proposed partition algorithms will be applied on both the model graphs and instances of a real life Market graph. Finally, the results for the market graph will be analysed further to interpret the structure of the market.

The paper is outlined as follows. The second chapter presents the necessary theoretical background. Its first section serves to introduce basic graph theory definitions and concepts. The following section covers graph partition and clustering. In Section 3, the theory of random graphs is presented along with the concepts of power law random graphs. Three different models for generating these graphs are discussed. The final section describes the Market graph model.

In Chapter 2, two different approaches for graph partitioning are presented, and formulated in terms of integer programs. Heuristic algorithms for computing both formulations are also introduced.

The forth chapter presents some empirical results from a case study of power law graph topologies. Properties and topological characteristics of graphs generated by the models introduced in Chapter 2 as well as instances of the Market graph model will be studied.

The main results are given in Chapter 5. Here, graph partitions for different graphs are presented. The approaches are tested on both simulated graphs and the market graphs and evaluated in terms of the quality of the obtained solutions. Specific focus will be put on studying the partition structure of the market graph. Finally, the partition of the market graph will be studied for several consecutive periods to study the dynamics and stability of the partition in the graph.

The final chapter includes conclusions and a discussion about open questions and possible directions of future work.

2

Theoretical background

2.1 Graph theory concepts

2.1.1 Basic definitions and notations

Since networks are represented in terms of graphs some notations from basic graph theory is introduced. Let $G = (V,E)$ be an undirected graph consisting of the set V with $|V| = n$ vertices and the set E of $|E| = m$ edges. We say that A_G is the adjacency matrix representing $G(V,E)$, if A_G is a $n \times n$ -matrix such that $A_G = [a_{ij}]_{i,j}^{n}$, with $a_{ij} = 1$ if $(i,j) \in E$ and $i \neq j$ and otherwise $a_{ij} = 0$. The *degree* d_i of a vertex i is the number of edges emanating from it. For every $d_i = d$, we can define $n(d)$ as the number of nodes in G with degree d. This give rise to a degree distribution of a graph G as the fraction of vertices having degree d. The *(open) neighbourhood* $T(i)$ of a vertex $i \in G$ is the set of all vertices sharing an edge with i, i.e $T(i) = \{j|a_{ij} = 1\}$. A *path* in G is a sequence of edges connecting vertices. The *average path length* is the average number of steps along the shortest path for all possible pairs of the network nodes. The *diameter* of the graph is the longest of all the shortest paths in the graph. The graph G is *connected* if there is a path from any vertex $v \in V$, to any vertex $u \in V$. We call G a *complete graph* if there exists an edge $(i,j) \in E$ for every $i \neq j$ and $i,j \in V$. Given a subset $S \subseteq V$, we denote by $G(S)$ the *subgraph* induced by the set S.

The complementary graph of G, denoted $\bar{G} = (V,\bar{E})$ is defined as follows. If $(i,j) \in E$ then $(i,j) \notin \bar{E}$ and if $(i,j) \notin E$ then $(i,j) \notin \bar{E}$. In words, one obtains the complementary graph of G by removing all the edges (i,j) present in G, and then introducing all the edges not present in G in the graph. The *edge density*, $\delta(G)$ measures the connectivity in the graph, defined by the ratio between the number of edges in the graph and the maximal possible number of edges in the graph. Mathematically we write

$$\delta(G) = \frac{2|E|}{|V|(|V|-1)}. \tag{2.1}$$

The *cluster coefficient* reveals to what extent the nodes in the graph tend to cluster together. The local clustering coefficient C_i for a vertex i with degree $d_i > 1$ is defined as the ratio of the number of edges among its neighbours divided by the maximal (possible) number of such edges. For $d_i \leq 1$ C_i is undefined. Mathematically we write C_i as

$$C_i = \frac{2E_i}{d_i(d_i-1)}, \quad d_i > 1 \tag{2.2}$$

where d_i is the degree of node i and E_i is the number of common edges among its neighbours. The global clustering coefficient C of the entire graph is defined as the mean of the local clustering coefficients, i.e., $C = \frac{1}{n}\sum_i^n C_i$.

2.1.2 Clusters, cliques and independent sets

Generally speaking, a *cluster* in a network is a set of elements that are more similar to each other than to elements not included in the cluster. Studying graph clusters can reveal topological structure of the network as well as information about the particular elements in the clusters. The similarity criterion varies depending on what property the cluster should reveal. Common criterias include vertex degree, vertex distance, or cluster density.

One special case of cluster called a *clique* is displayed in Figure 2.1. We say that $C \subseteq V$ is a clique if the induced sub-graph $G(C)$ is complete. A clique is *maximal* if it cannot be contained in any larger clique in the graph, and it is called a *maximum clique* if it is a clique of maximal cardinality in the graph. A problem in graph theory is to identify maximum cliques in a graph, called the *maximum clique problem*, (MC.). The size of a maximum clique is called the *clique number*, denoted $\omega(G)$.

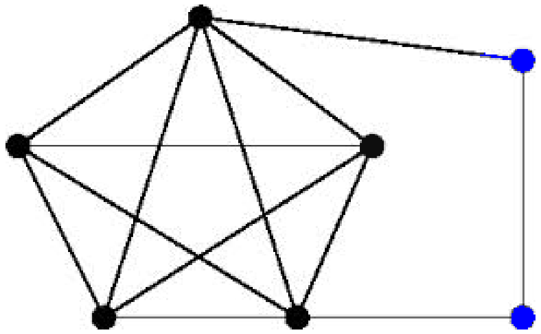


Figure 2.1: Example of a graph $G(C)$, induced by the clique C (black nodes)

Since the strict requirements of cohesiveness in the clique definition often is difficult to fulfill, several relaxations of cliques have been introduced. Examples of clusters being cliques relaxations include *k-clubs*, *k-cores*, *k-communities* and γ *-quasi clique*, all further discussed in [4]. We say that the set $Q \subseteq V$ with $|Q| = p$ is a γ *-quasi clique*, $(0 < \gamma < 1)$ if the graph $G(Q)$ induced by Q is connected and satisfies $|E(G(Q))| \geq \gamma\binom{p}{2}$. This means that we impose the requirement that the edge density of the induced graph $G(Q)$ must be greater or equal to the threshold γ. Note that in the case when $\gamma = 1$, then Q corresponds to a clique.

The opposite of a clique is an *independent set.* An independent set is a set $I \subseteq V$ such that the induced graph $G(I)$ has no edges. The problem of finding an independent set of maximal cardinality in a graph is called the *maximum independent set problem* (MIS.). By $\alpha(G)$ we denote the size of the largest independent set of G. Note the symmetry between the maximum clique problem and the maximum independent set problem. The set Q is a maximum clique in $\bar{G}$ if and only if Q is a maximum independent set in G. Therefore a MIS. can easily be reformulated into a MC. and vice verse, and hence it holds that $\omega(G) = \alpha(\bar{G})$.

2.2 Clustering and graph partitions

Clustering involves the task of partitioning the elements of the graph into disjoint clusters. Generally one seeks a partition of the vertices in a way that maximizes the similarity within the clusters and minimizes the similarity between the clusters. A partition where each cluster is a clique is called a *clique partition.* The *minimal clique partition problem* is to find the smallest integer k such that the vertex set V of G can be partitioned into the k disjoint sets C_1,...,C_k, where each C_i is a clique. This minimal integer k is called the clique partitioning number $\bar{\chi}(G)$.

A concept closely related to graph partitioning is graph coloring. A proper *k-coloring* of the vertices of G is an assignment of colors to the vertices in G such that no adjacent vertices in G have the same color. If such a coloring exists we call the graph G *k-colorable.* Seeking a coloring using a minimal number of colors is called the *graph coloring problem.* The smallest integer k for which the graph G is k-colorable is the *chromatic number* of G denoted $\chi(G)$. In a coloring of G the vertices with the same color are all pairwise non-adjacent, making them by definition independent sets. Thus, the graph coloring problem is equivalent to finding a minimal partition of G into pairwise, disjoint independent sets. Due to the symmetry between cliques and independent sets the *graph coloring problem* of $\bar{G}$ can therefore also be formulated as the *minimum clique partition problem* of G. Again, due to the symmetry we have that $\bar{\chi}(G) = \chi(\bar{G})$.

2.2.1 Desirable cluster properties

What constitutes a cluster of high quality will of course depend on the application at hand. However, some characteristics are relevant for most structures. First, the cluster must be connected, thus if there is no path between two vertices u, and v, they should not be grouped within the same cluster. By classifying edges as internal if they connect vertices within a cluster to each other, the *internal degree* of a vertex v in a cluster $C \subset V$ as $deg_{int}(v,C) = |T(v) \cap C|$, where $T(v)$ is the neighbourhood of v in G. Similarly, edges are identified as external if they connect a vertex in a cluster with a vertex outside the cluster. Thus the *external degree* of a vertex v in a cluster C is $deg_{ext}(v,C) = |T(v) \cap (V \setminus C)|$. Note that with these definitions we have $d_v = deg_{int}(v,C) + deg_{ext}(v,C)$.

In general, if $deg_{int}(v,C) = 0$, then v should not be included in cluster C as it is not connected to the other vertices in C. Similarly $deg_{ext}(v,C) = 0$ implies that C could be a good cluster for v as it has no connections outside C. Generally in clustering one seeks to form clusters such that the induced sub-graph is dense and has few connections to the rest of the graph. We therefore introduce two density measures with respect to a cluster C. We call the density of the sub-graph induced by C *internal* or *intra-cluster density* if it is defined by

$$\delta_{int}(C) = \frac{|\{(u,v) \in E | v \in C, u \in C\}|}{|C|(|C|-1)} = \frac{1}{|C|(|C|-1)} \sum_{v \in C} deg_{int}(v,C). \tag{2.3}$$

Given a clustering of a graph G into k clusters $\bar{C} = (C_1,C_2...,C_k)$ we define the *intra-cluster density* of the clustering $\bar{C}$ as the average of the intra-cluster densities of the included clusters.

$$\delta_{int}(G|C_1,C_2...,C_k) = \frac{1}{k} \sum_{i=1}^{k} \delta_{int}(C_i). \tag{2.4}$$

Similarly, we introduce the *external* or *iter-cluster density* of a clustering as the ratio of the number of external edges and the maximal possible number of external edges.

$$\delta_{ext}(G|\{C_1,C_2...,C_k) = \frac{|(u,v)|v \in C_i, u \in C_j, i \neq j\}|}{n(n-1) - \sum_{l=1}^{k} |C_l|(|C_l|-1)} \tag{2.5}$$

Employing the introduced density measures above a good clustering should have an internal density significantly higher than that of the overall graph, $\delta(G)$ and an external density much lower than $\delta(G)$. Depending on how strict these density constraints are imposed different cluster types can be obtained. The loosest possible definition being a connected component and the strictest being a maximal clique. However, in practice most interesting structures can be found somewhere in between. Computation of connected components can be done in $O(n+m)$ time with a breadth-first search while identifying maximal cliques is NP-complete [5].

2.2.2 Clustering structure

An important characteristic in a clustering structure is whether the clusters C_1,C_2...,C_k must be disjoint or if cluster overlap is allowed. In the former case we talk about a *graph partition*, or a ”hard” clustering where $C_i \cap C_j = \emptyset$, $\forall i \neq j$. When clusters overlap, we call this a *graph cover* of a ”soft” clustering. In this paper we will focus on the former structure and we will use the term clustering and partition exchangeable, always referring to the hard clustering.

Another distinction for a clustering structure is the one between *flat* versus *hierarchical* clustering. If the partition consists of a set of clusters without any explicit structure that would relate clusters to each other we talk about a flat clustering. On the other hand, we say that a clustering is hierarchical if it contains several levels of clusters where each top level cluster consists of clusters from lower levels. This way the clusters can be represented in terms of a tree structure, called a dendrogram, Figure 2.2 shows an hierarchical clustering with its corresponding dendrogram. Which type of clustering that is preferred depends on the network topology. If it is known that the data contains a hierarchical structure, then this should be preferred. However, if the number of clusters are known prior, then a flat clustering approach is preferred over a hierarchical structure, [5].

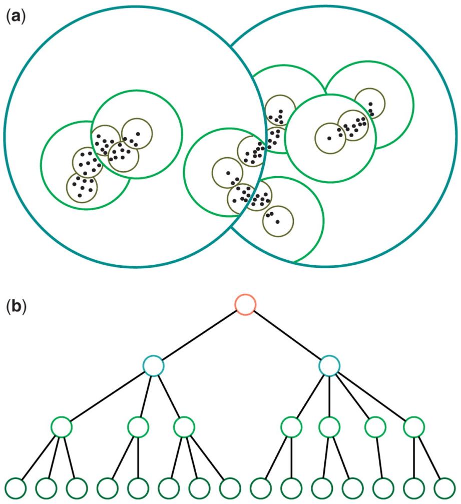


Figure 2.2: An hierarchical clustering, represented by (a) set division, (b) dendrogram. [6]

Hierarchical clustering can be separated further into two types, depending on whether the partition is refined or coarsened between each level. In the first type, called top-down or *divisive* hierarchical clustering the graph is recursively spilt into smaller and smaller pieces. In the second version, bottom-up or *agglomerative* clustering, smaller clusters are iteratively merged into larger ones.

2.2.3 Measures to identify clusters

Clusters are usually identified with two different approaches, using vertex similarities or a fitness measure. In the former approach one computes a set of similarity values for all vertices and then classifies them into clusters according to their overall score. In the latter case one computes a fitness function over the set of possible clusters and then chooses among the set of clusters that optimize the chosen fitness measure. An extensive overview of clustering techniques can be found in [5, 7].

Density based measures

Some approaches uses a density based fitness measure to identify maximal sub-graphs with a density higher than a certain threshold. As Schaeffer [5] mentions, finding clusters based on their edge-density can essentially be considered as special cases of the following decision problem:

Instance: *Given an undirected graph $G = (V,E)$, with a density measure $\delta(\cdot)$ over the vertex subsets $S \subseteq V$, a positive integer $k \leq |V|$ and a rational number $\xi \in [0,1]$.*

Question: *Does it exist a subset $S \subseteq V$ such that $|S| = k$ and the density $\delta(S) \geq \xi$?*

Note that if the density measure used is the overall graph density the problem is NP-complete since for $\xi = 1$ it coincides with the NP-complete maximum clique problem. Many variants and relaxations of this problem have been proposed and studied during the years. Matsuda et al. proposed a model that considers γ-quasi cliques as clusters [8]. They showed that it is NP-complete to determine whether a given graph has a $\frac{1}{2}$ quasi clique of order at least k.

Cut based measures

Instead of focusing on the internal density of the cluster one can also measure how connected the cluster is to the rest of the graph. These measures are usually based on cut sizes. Given a graph $G = (V,E)$ and two subsets $S_1 \subseteq V$, $S_2 \subseteq V$ we define the *cut size*, $c(S_1, S_2)$ of S as the number of edges between nodes in S_1 and nodes in S_2. Mathematically, we write this as

$$c(S_1,S_2) = |\{(u,v) \in E | u \in S_1, v \in S_2\}|. \tag{2.6}$$

The definition in (2.6) can be extended to a collection of clusters $\Pi = (V_1,....,V_K)$ as the sum of all edges with end nodes in different clusters. We define the cut of a collection of clusters $\Pi = (V_1,..., V_K)$ as

$$C(\Pi) := \frac{1}{2}\sum_{i=1}^{K} c(V_i,\bar{V_i}) \tag{2.7}$$

where $\bar{V_i}$ is the complement of V_i in V and $c(V_i,\bar{V_i})$ is given by (2.6) and as before, $\bar{V_i} = V \setminus V_i$.

If the cut is normalized by the sizes of the corresponding clusters, we get the *Ratio Cut*, $C_R(\Pi)$ defined as

$$C_R(\Pi) := \frac{1}{2}\sum_{i=1}^{K} \frac{c(V_i,\bar{V_i})}{|V_i|}. \tag{2.8}$$

Another normalization was introduced by Shi and Malik [9], called the *Normalized cut*, $C_N(\Pi)$. They defined it as the ratio between the cut size and the degrees of the vertices.

$$C_N(\Pi) := \frac{1}{2}\sum_{i=1}^{K} \frac{c(V_i,\bar{V_i})}{vol(V_i)} \tag{2.9}$$

where $vol(V_i) = \sum_{j \in V_i} d_j$, i.e. the sum over the degrees of the vertices in V_i.

Modularity

Another common measure to identify graph clusters is the metric *modularity,* introduced by Newman and Girvan in [10]. The metric modularity, denoted Q, is defined as

$Q(\Pi) =$ (*the number of the edges that fall within a cluster*) - (*the expected such number if edges were distributed at random*)

The meaning of the first term is clear. However, the second term requires some comments. Determine the expected number of edges in a cluster necessitate choosing a null model for the network, a question we will address soon. First, we introduce P_{ij} as the probability that there is an edge between vertex i and j. Thus, the actual, minus the expected number of edges between i and j can be written $A_{ij} - P_{ij}$ and the modularity is proportional to the sum of this quantity over all pairs of vertices in the same cluster. Thus, the modularity can be expressed as

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - P_{ij}]\delta(C_i,C_j) \tag{2.10}$$

where $\delta(C_i,C_j) = 1$ if $C_i = C_j$ and zero otherwise.

Returning to the question of choosing a null model. A possible choice could be to consider a standard uniform random graph, in which edges appear random with equal probability $P_{ij} = p$. However, this model turns out to be a bad representation for many real life graphs. In particular the model often fails to reflect the degree distribution of the graph. One way to deal with this in practice is to approximate the expected degree of each vertex within the model with the actual degree, d_i of the corresponding vertex i in the real network. The expected degree of i is given by $\sum_j P_{ij}$, giving us the relation

$$\sum_j P_{ij} = d_i \tag{2.11}$$

The simplest null model in this class, is the one in which edges are distributed at random subject to the constraint (2.11). This implies that the expected number of edges between i and j, P_{ij} can be expressed as a product of separate functions of the degrees.

$$\sum_j P_{ij} = f(d_i)\sum_j f(d_j) = d_i$$

Hence, $f(d_i) = Cd_i$, for some constant C. Furthermore, since $\sum_i d_i = 2m$, (m being the number of edges in the graph) we can write

$$2m = \sum_i\sum_j P_{ij} = C^2\sum_i\sum_j d_i d_j = (2mC)^2$$

which gives $C = \frac{1}{\sqrt{2m}}$, and hence $P_{i,j} = \frac{d_i d_j}{2m}$.

Thus, the modularity (2.10) can be rewritten as

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{d_i d_j}{2m}\right]\delta(C_i,C_j) \tag{2.12}$$

2.3 Random graph theory

2.3.1 Uniform random graphs

The theory of random graphs was introduced in 1959 in the work of Erdös and Renyi [11]. In the context of their probabilistic method a random graph can be described in the following way. Consider the situation where we try to study the existence of graphs G_P with a specific property P. Let the existence of such a graph be represented by the random variable X. Then, one can construct a probability space such that the appearance of G_P with property P can be described by the event E. Showing that the probability of observing this event E is larger than zero, i.e. showing that $P(X = E) > 0$ implies that such a graph G_P with property P in fact can exist. By studying the distributions of probability spaces of this kind random graphs are introduced.

In their first paper Erdös and Renyi introduced two formulations for the uniform random graph model. The first version, $G(n,m)$ assigns a uniform probability to all graphs with n nodes and m edges. By setting $N = \binom{n}{2}$ we can see that $G(n,m)$ has $\binom{N}{m}$ elements, all with probability $\binom{N}{m}^{-1}$. In the second formulation denoted $G(n,p)$, a graph is constructed by introducing edges between nodes with an independent probability p, where $0 < p < 1$. One can easily identify similarities between the two formulations since all graphs with n nodes and m edges will have the same probability $p^m(1-p)^{\binom{n}{2}-m}$ in the $G(n,p)$ model. From now on we will continue working with the second formulation of the model.

With the notation above a graph in $G(n,p)$ has $\binom{n}{2} \cdot p$ expected number of edges. Therefore the degree distribution of a particular vertex v is given by the *Binomial distribution*, and we have

$$P(d_v = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}. \tag{2.13}$$

Letting $n \to \infty$ we get that for the case $np = constant$ the degree distribution tends to the *Poisson distribution*, [12].

$$P(d_v = k) = \frac{(np)^k e^{-np}}{k!} \tag{2.14}$$

Many properties of the $G(n,p)$ model have been studied, some fundamental results cover graph connectivity, emergence of a giant connected component, as well as results about graph diameter, independent sets, cliques and colorings. The interested reader is referred to [12] for a more comprehensive review of the different properties of random graphs. Proving the existence of many of these properties rely on studying the probability space as n tends to infinity. One says that the random graph $G(n,p)$ *asymptotically almost surely, (a.a.s)* has a property Q if $\lim_{n\to\infty} P[G(n,p) = Q] = 1$. Many graph properties undergo structural changes as the edge density passes some limit [13]. As this limit is passed a graph undergoes a *phase transition* from not having the property Q to having the property Q. This is referred to as the *threshold function* of the property Q. A *threshold function* $r(n)$ is defined as:

$r(n)$ is called a *threshold function* for a graph theoretic property Q if

(i) When $p(n) << r(n)$, $\lim_{n\to\infty} Pr[G(n,p) = A] = 0$

(ii) When $p(n) >> r(n)$, $\lim_{n\to\infty} Pr[G(n,p) = A] = 1$

In words this means that $p(n) << r(n)$ implies that $G(n,p)$ does not have property Q and $p(n) >> r(n)$ implies it does have property Q. If such a threshold function exists for a property we say that a phase transition occurs at the threshold. The observation of such phase transitions was one of the main contributions of [11].

Two characteristics worth mentioning in this context is the degree distribution and cluster coefficient of a random graph $G(n,p)$. First, as stated above the degree distribution for $G(n,p)$ tends to the Poisson distribution as n grows large. This is the first drawback when using this model to represent real-life graphs. Many real life graphs have instead shown to exhibit a degree distribution with a heavy right end tail [14, 15]. This kind of degree distribution is often referred to as a *power law distribution*. Secondly, the clustering coefficient of a random graph $G(n,p)$ is given by, $C_R = \frac{<k>}{n} = p$. This is a second indication that $G(n,p)$ is not suitable for modelling real life networks since it has been shown that in many real life graphs the clustering coefficient highly exceeds this number, [16].

2.3.2 Power law random graphs

Following the discoveries that the topology of many real life networks could not be accurately modelled by the classical uniform random graph theory new models for describing these scale free networks have been presented. A common feature of these models is the occurrence of a power law in the right end tail of their degree distribution. This section will therefore introduce the power law distribution and its specific properties. We then move on and discuss some proposed graph models for generating networks with a power law degree distribution.

Power law distribution

One says that the random variable $X > 0$ follows a *power law* if it has the probability density function.

$$f(x)_X = \frac{\alpha}{x^{\beta}}, \quad x \in S \tag{2.15}$$

where S is the support of x, α is a normalization constant and β is the power law exponent. Note that by taking the logarithm of both sides the relationship (2.15) will be linear in a log-log scale with coefficient $-\beta$ and intersection $\log(\alpha)$. A common power law distribution is the *Pareto distribution*, defined as

$$f(x)_X = \begin{cases} \dfrac{\beta x_{min}^{\beta}}{x^{\beta+1}}, & x \geq x_{min} \\ 0 & x < x_{min} \end{cases} \tag{2.16}$$

The corresponding discrete distribution is called the *Zipf distribution*.

Scale invariance

A characteristic of power law distributions is their scale invariance property. That a function $f(x)$ is scale invariant means that scaling x with a constant c is equivalent to scaling the function itself with a constant, that is:

$$f(x) = \alpha x^{\beta} \Rightarrow f(cx) = \alpha(cx)^{\beta} = c^{\beta} f(x) \tag{2.17}$$

Moments

Another topic worth mentioning about power law distributions is the limited existence of higher moments. The *k:th moment* of a probability distribution is defined as

$$< x^k >= \int_{-\infty}^{\infty} x^k p(x) dx \tag{2.18}$$

With for example the Pareto distribution defined in (2.16), we get the *k:th* moment as

$$< x^k >= \beta x_{min}^{\beta} \int_{x_{min}}^{\infty} x^{k-\beta+1} dx \tag{2.19}$$

We can see that for $k = 1$, corresponding to the mean, the integral (2.19) will diverge for $1 < \beta \leq 2$. When $2 < \beta \leq 3$ the mean will be finite but the second moment (variance) will still be infinite. Only for $\beta > 3$ the distribution will have both finite mean and variance.

The concept of a *power law graph* arises when the degree distribution of the vertices in a graph G follows (or closely approximates) some power law, i.e., when the number of vertices y with degree x in the graph can be described by the relation $y = \frac{e^{\alpha}}{x^{\beta}}$. A more precise, universal definition is not available, but must be specified within the particular graph model.

2.3.3 Models for generating power law random graphs

Several models for generating random graphs with a topology such that their degree distribution follow a power law have been developed and analyzed in recent years. Since this feature was first observed in graphs representing real life networks the developed models often try to mimic the topology of these specific graphs. As a consequence the different models all create graphs with a degree distribution approximating a power law, however they differ in many other topological characteristics, such as edge density, clustering coefficient, and average path length. This is partly due to the fact that there is still no strict universal, mathematical definition of what constitutes a power law graph. Usually graph models can be divided into two different groups, *curve fitting generators* and *preferential attachment generators.*

Curve fitting generators make use of an explicit, scale free degree distribution $D = (d_1,d_2,...,d_N)$ to connect N nodes in such a way that the resulting graph G has the desired degree distribution D. The family of preferential attachment generators combines the idea of network growth with preferential attachment of the vertices. Starting with a small connected graph the growth of the network is divided into time steps in which the probability that a new edge will be connected to a vertex in the graph is proportional to the degree of the vertex. For an extensive review of developed generators the reader is referred to [17]. We will focus on three different, well known models. First the Power Law Random Graph model (PLRG) belonging to the curve fitting family and later the Albert-Barabasi (BA) and the Copying model (COPY), belonging to the second family of generators.

Power Law Random Graph

The *Power Law Random Graph* is due to Aiello, Chung and Lu [18]. The model denoted by $G(\alpha, \beta)$ assigns uniform probability to all graphs $G = (V,E)$ with a degree distribution satisfying;

$$P(|v \in V|deg(v) = x|) = y = \left[\frac{e^{\alpha}}{x^{\beta}}\right] \tag{2.20}$$

where y is the number of vertices with degree x. The $[\cdot]$ in (2.20) refers to the integer part of $\frac{e^{\alpha}}{x^{\beta}}$. This is necessary since vertex degrees can only take integer values. An assumption in the model is that the sum of all degrees in the graph must be even, the motivation for this will be clear later. In this formulation the maximal possible node degree in the graph is equal to $e^{\frac{\alpha}{\beta}}$. By summing the density function over all possible degrees one can express the number of vertices in the model as

$$N = \sum_{x=1}^{e^{\frac{\alpha}{\beta}}} \frac{e^{\alpha}}{x^{\beta}} = \begin{cases} \zeta(\beta)e^{\alpha}, & \beta > 1 \\ \alpha e^{\alpha} & \beta = 1 \\ \dfrac{e^{\frac{\alpha}{\beta}}}{1-\beta} & 0 < \beta < 1 \end{cases} \tag{2.21}$$

where $\zeta(t) = \sum_{n=1}^{\infty} \frac{1}{n^t}$ is the *Riemann zeta function.*

The expected number of edges in the graph can be computed by

$$E = \frac{1}{2} \sum_{x=1}^{e^{\frac{\alpha}{\beta}}} x \frac{e^{\alpha}}{x^{\beta}} = \begin{cases} \dfrac{1}{2}\zeta(\beta - 1)e^{\alpha}, & \beta > 2 \\ \dfrac{1}{4}\alpha e^{\alpha} & \beta = 2 \\ \dfrac{1}{2}\dfrac{e^{\frac{2\alpha}{\beta}}}{2-\beta} & 0 < \beta < 2 \end{cases} \tag{2.22}$$

The explicit construction of a graph can be described as follows. A degree sequence $D = (d_1,d_2,...,d_N)$ is drawn from a truncated Pareto distribution with the input values, the target number of nodes N and a power law exponent, β. Note that these values will uniquely determine the scaling constant α. The degree sequence is then assigned to the N nodes in the graph. Then, for each node i we create d_i "stubs" (can be considered as half edges which needs to be connected to another half). The number of "stubs " is even since it is equal to the sum of the degrees in the graph. Now, every "stub" will be connected to another one, chosen at random and without repetition. Due to the random choice in the matching the resulting graph may not be connected, and can include self-loops and duplicating links. However, by adding a post processing that eliminates self loops and disconnected components a connected, simple graph can be obtained. The procedure is not exact but will asymptotically yield power law graphs [18]. For a given degree sequence D the procedure can be described by Algorithm 1.

The authors in [19] showed several characteristics of the model, including the following proposition.

Proposition 2.3.1 *For* $2 < \beta < \beta_0 = 3.47875$ *the random graph* $G(\alpha, \beta)$ a.a.s *has a unique giant connected component, and the size of the second largest component is of size* $O(log(N))$.

Barabasi-Albert Model

The model introduced by Barabasi and Albert [20] is based on preferential attachment and network growth. The algorithm starts with a small, complete graph of size m_0 and adds in each time step one vertex with m edges to the graph. The probability that a

Algorithm 1: PLRG generator

```
Input: Degree sequence, $D = (d_1,...,d_N)$
Result: Edge list $E$ for graph $G$
Initialize $E = [\ ]$ ;
for $j = 1 : N$ do
    $E = [E; j \cdot ones(d_j,1)]$ ;
end
$M = length(E)$ ;
randomize the position of the rows in $E$ ;
for $j = 1 : M/2$ do
    connect $E(j)$ to $E(M - j + 1)$ ;
end
```

new vertex will be adjacent to vertex i in the graph is proportional to the degree of the latter, d_i, such that:

$$P(X = i) = \frac{d_i(t)}{\sum_{\forall j} d_j(t)} \tag{2.23}$$

This relation describes the preferential attachment for high degree nodes of the model. The concept is sometimes refereed to as the "richer get richer" phenomena. Using a continuum theory approach as in [16] it can be proved that this model will generate a power law graph topology. By considering the degree d_i of a node i as a continuous real variable one finds that the rate at which the d_i changes will be proportional to (2.23), and d_i will therefore satisfy

$$\frac{\partial d_i}{\partial t} = m \cdot P(X = i) = m \cdot \frac{d_i(t)}{\sum_{\forall j} d_j(t)} \tag{2.24}$$

using that $\sum_{j=1}^{N-1} d_j = 2mt - m$ at time t this can be rewritten as

$$\frac{\partial d_i}{\partial t} = \frac{d_i}{2t - 1} \tag{2.25}$$

For large t, we can neglect the -1 in the denominator, giving us

$$\frac{\partial d_i}{d_i} = \frac{1}{2}\frac{\partial t}{t} \tag{2.26}$$

By integrating of (2.26) and using that all vertices have initial degree $d_i(t_i) = m$ the solution for the (2.26)

$$d_i(t) = m\left(\frac{t}{t_i}\right)^{1/2} \tag{2.27}$$

Using (2.27), the probability that a node i has degree $d_i < d$ can be expressed by

$$P[d_i(t) < d] = P(t_i > \frac{m^2 t}{d^2}) \tag{2.28}$$

Assuming that the growth process is divided into equal time intervals the t_i values will have a constant probability density, $P(t_i = \frac{1}{m_0+t})$. Substituting this into (2.28) we get that

$$P\left(t_i > \frac{m^2 t}{d^2}\right) = 1 - P\left(t_i \leq \frac{m^2 t}{d^2}\right) = 1 - \frac{m^2 t}{d^2(t+m_0)} \tag{2.29}$$

Finally, the probability density function can be obtained using that

$$P(d) = \frac{\partial P[d_i(t) < d]}{\partial d} = \frac{2m^2 t}{(m_0+t)}\frac{1}{d^3} \tag{2.30}$$

Thus, the BA model will generate a graph with a power law degree distribution, with power law exponent equal to $\beta = 3$ independent of the parameters m and m_0. However, one can note that the scaling constant of the distribution will be proportional to m^2.

In practice the process of creating a graph can be described by the pseudo code in Algorithm 2.

Copying model

The copying model, (COPY) was first introduced by Kleinberg, Kumar, Raghavan, Rajagopalan and Tomkins in [21] and [22] to model the characteristics of the Web graph. Like the BA model it is based on network growth, however the attachment process differs. The basic mechanism can be described as follows. First the graph is initialized by a small clique. Then, for every new vertex v, introduced in the graph, a single vertex u, is chosen uniformly at random from the graph nodes. For each neighbour u_i of u connect u_i and v with probability q and with probability $1-q$ connect v with a random vertex. A result of this is that $d_v = d_u$. The first process where neighbours of u are connected to v increases the probability of high-degree vertices receiving new incoming edges. Hence, this part corresponds to the preferential attachment procedure in the BA model.

The evolution of the node degrees in the graph can be found by considering the following case. The degree of an existing vertex j could increase in two ways. First, if one of the

Algorithm 2: Barabasi- Albert generator

```
Input: Number of nodes $N$, edges to attach $m$
Result: Edge list $E$ for graph $G$
First step: create clique with $m_0$ nodes ;
$[E_{core}]$ = CreateCore($m_0$) ;
$E = [E_{core}]$ ;
$degree = [m_0 - 1 * ones(m_0,1); zeros(N - m_0,1)]$ ;
Second step: attach remaining nodes with preferential attachment bias. ;
for $i = m_0 + 1...N$ do
    $p_{cum} = cumsum(degree(1 : i - 1))./sum(degree(1 : i - 1)$ ;
    $nodes_{chosen} = zeros(1,m)$ ;
    $r = random(1,m)$ ;
    for $j = 1 : m$ do
        $nodes_{chosen}(1,j) = min(find(r(1,j) < p_{cum})$ ;
    end
    $nodes_{chosen} = unique(nodes_{chosen})$ ;
    Create reciprocal edge between $i$ and $nodes_{chosen}$ in $E$ ;
    Update $degree$ vector for node $i$ and $nodes_{chosen}$ ;
end
```

neighbours of j is copied by the new vertex, j will increase its degree with probability q. Alternatively j could be chosen directly from uniform attachment. Assuming we want to create a network with N nodes, we will have a random process with N steps. Let the random variable $X_j(t)$ represent the number of in-links to vertex j at time $t \geq j$. Using the initial condition $X_j(j) = 0$ (has no in-links when introduced) and assuming that every introduced vertex has initial degree 1 we can write the probability that node $t + 1$ links to node j, (i.e. that vertex j increases its in degree by one) as

$$\frac{p}{t} + \frac{qX_j(t)}{t} \tag{2.31}$$

By approximating the discrete random variable $X_j(t)$ with a continuous function of time $x_j(t)$ we can write

$$\frac{dx_j}{dt} = \frac{p}{t} + \frac{qx_j(t)}{t} \implies \frac{1}{p + qx_j}\frac{dx_j}{dt} = \frac{1}{t} \tag{2.32}$$

Integrating (2.32) gives us

$$\int \frac{1}{p + qx_j}\frac{dx_j}{dt}\mathrm{d}t = \int \frac{1}{t}\mathrm{d}t$$

$$\begin{aligned}
&\Longrightarrow ln(p+qx_j) = qln(t) + c, \quad \text{setting} \quad A = e^c \\
&\Longrightarrow p + qx_j = At^q, \\
&\Longrightarrow x_j(t) = \frac{1}{q}(At^q - p) \qquad (2.33)
\end{aligned}$$

Using the initial condition, $x_j(j) = 0$ we get

$$0 = x_j(j) = \frac{1}{q}(Aj^q - p) \Rightarrow A = \frac{p}{j^q}$$

And thus we have $x_j(t)$ as

$$x_j(t) = \frac{1}{q}(\frac{p}{j^q} \cdot t^q - p) = \frac{p}{q}\left[\left(\frac{t}{j}\right)^q - 1\right] \qquad (2.34)$$

Now, for a given value k, and time t we look for the fraction of vertices in the graph with at least k in-links. Using the continuous approximation we look for the fraction of functions satisfying $x_j(t) \geq k$, giving us

$$x_j(t) = \frac{p}{q}\left[\left(\frac{t}{j}\right)^q - 1\right] \geq k \Rightarrow j \leq t\left[\frac{q}{p} \cdot k + 1\right]^{-\frac{1}{q}}$$

Thus, the fraction of values j (out of the total t) that will satisfy this is

$$\frac{1}{t} \cdot t\left[\frac{q}{p} \cdot k + 1\right]^{-\frac{1}{q}} = \left[\frac{q}{p} \cdot k + 1\right]^{-\frac{1}{q}}$$

This approximates the number of nodes with at least k in-links, which we will denote by $F(k)$. Finding the number of nodes with exactly k in-links can be obtained by differentiating $F(k)$, $f(k) = \frac{dF(k)}{dk}$, giving us

$$f(k) = \frac{1}{q}\frac{q}{p}\left[\frac{q}{p} \cdot k + 1\right]^{-1-\frac{1}{q}}$$

Thus, the fraction of nodes $f(k)$ with k in-links is proportional to $k^{-(1+\frac{1}{q})}$. Since, $q \in$ [0,1] we can see that the power law exponent of $f(k)$, α can take values between $[2, \infty]$.

The explicit procedure of creating a graph can be described by Algorithm 3.

Many more elaborate versions have been developed from this basic mechanism, for example the model *Forestfire*, [23].

Algorithm 3: COPY model generator

```
Input: Number of nodes $N$, copy threshold probability $q$, initial clique size $m$.
Result: Edge list $E$ for graph $G$
First step: create clique of size $m$ ;
CreateCore($m$) ;
Second step: attach remaining nodes through copying mechanism. ;
for $i = m + 1,...N$ do
    $u$ = random copy node selected from 1,...$i - 1$ ;
    $d_u = degree(u)$ ;
    $neigbour_u$ = vector with neigbours of $u$ ;
    for $j = 1 : d_u$ do
        select $r$ at random, $r \in U(0,1)$ ;
        if $r > q$ then
            Create reciprocal edge in $E$ between $i$ and $neighbour_u(j)$ ;
        end
        else
            Create reciprocal edge in $E$ between $i$ and vertex $t$ chosen at random
            from 1...,$i - 1$ ;
        end
    end
end
```

2.4 The Market graph model

A real life power law graph will also be considered. Employing the method introduced in Boginski, Butenko and Pardalos [2] we construct the market graph by representing traded instruments by vertices and introducing edges if the Pearson cross-correlation between two instruments exceeds a certain threshold, θ. This can be expressed in terms of the graph adjacency matrix $A = [a_{i,j}]_{i,j=1}^{n}$ as

$$a_{ij} = \begin{cases} 1, & \text{if } C_{i,j} \geq \theta \\ 0, & \text{if } C_{i,j} < \theta \end{cases} \tag{2.35}$$

where $\theta \in [-1,1]$. The cross correlation between i and j is given by

$$C_{i,j} = \frac{E(R_i R_j) - E(R_i)E(R_j)}{\sqrt{Var(R_i)Var(R_j)}} \tag{2.36}$$

where $R_i(t)$ is the daily return of instrument i at time t.

$$R_i(t) = \frac{P_i(t)}{P_i(t-1)} \tag{2.37}$$

and $P_i(t)$ is the closing price of instrument i at time t.

This results in an undirected, unweighted graph, represented with an adjacency matrix $A(\theta) = [a_{i,j}]_1^n$, where $a_{i,j}$ is 1 if there is an edge between i and j and 0 otherwise.

Graph characteristics such as edge distribution, cluster coefficient, maximum cliques and independent sets can be examined to study the structure of the market. Many previous studies have shown that above a certain threshold the degree distribution of the market graph will follow a power law, [1, 3, 24, 25].

In this paper we focus on the problem of partitioning the market graph into disjoint clusters. In terms of the market graph this can be interpreted as a division into different, strongly connected segments of the market.

3

Graph Partitioning Methods

3.1 Problem formulations

In this section we will present two formulations for graph partition based on two different fitness measures, the normalized cut, (2.9) and modularity, (2.12). Both formulations result in integer programs which turn out to be NP-hard problems.

3.1.1 Minimizing Normalized cut

The first formulation seeks a partition of V into (a fixed number of) k disjoint subsets such that the normalized cut (2.9) of the partition is minimized. This approach was introduced in [9] for image segmentation and is solved using a spectral relaxation of the problem. The approach was further studied in [26]. The objective function in this case is given by

$$\underset{(A_1,...,A_k)}{\text{minimize}} \quad C_N(A_1,...,A_k) \tag{3.1}$$

where $C_N(\cdot)$ is defined by (2.9).

We will first consider the case when $k = 2$, since the formulation is easiest to understand in this case. Hence, we seek a bisection $\Pi = (A, \bar{A})$ of V that minimizes (3.1)

First, we define the cluster indicator function f as

$$f_i = \begin{cases} \sqrt{\frac{vol(\bar{A})}{vol(A)}}, & \text{if } v_i \in A \\ -\sqrt{\frac{vol(A)}{vol(\bar{A})}}, & \text{if } v_i \in \bar{A} \end{cases} \tag{3.2}$$

where as before

$$vol(A) = \sum_{i \in A} d_i. \tag{3.3}$$

Let D be the matrix with the node degrees on the diagonal, $D = diag(d_1,..., d_n)$. Then, we have that

$$(Df)'\mathbf{1} = \sum_{i=1}^{n} d_i f_i = 0, \tag{3.4}$$

and

$$\begin{aligned} f'Df &= \sum_{i=1}^{n} d_i f_i^2 \\ &= \frac{vol(\bar{A})}{vol(A)} \sum_{i \in A} d_i + \frac{vol(A)}{vol(\bar{A})} \sum_{i \in \bar{A}} d_i \\ \text{using (3.3)} &= vol(A) + vol(\bar{A}) \\ &= vol(V). \end{aligned} \tag{3.5}$$

Now, let $L = D - A$ be the Laplacian matrix, defined as in Appendix B. Then, using Proposition 8.1.1 we can write

$$\begin{aligned} f'Lf &= \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(f_i - f_j)^2 \\ &= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} a_{ij} \left(\sqrt{\frac{vol(\bar{A})}{vol(A)}} + \sqrt{\frac{vol(A)}{vol(\bar{A})}} \right)^2 + \frac{1}{2} \sum_{j \in A, i \in \bar{A}} a_{ij} \left(-\sqrt{\frac{vol(\bar{A})}{vol(A)}} - \sqrt{\frac{vol(A)}{vol(\bar{A})}} \right)^2 \\ &= vol(V) C_N(A,\bar{A}). \end{aligned} \tag{3.6}$$

Thus, for $k = 2$ the problem (3.1) can be rewritten as

$$\begin{aligned} \underset{A}{\text{minimize}} \quad & f'Lf \\ \text{subject to} \quad & f \text{ as in } \quad (3.2) \\ & \mathbf{1}'Df = 0 \\ & f'Df = vol(V). \end{aligned} \tag{3.7}$$

For each f_i there is 2 possible choices, depending on if v_i belongs to A or not. In [9] the authors showed that (3.7) is NP-complete even for a regular grid. A possible relaxation is to discard the condition of discreteness and allow f_i to take arbitrary values in $\mathbf{R}$. Imposing this relaxation leads to the following relaxed problem

$$\begin{aligned} \underset{f \in \mathbf{R}^n}{\text{minimize}} \quad & f'Lf \\ \text{subject to} \quad & \mathbf{1}'Df = 0 \\ & f'Df = vol(V). \end{aligned} \tag{3.8}$$

By introducing, $g := D^{1/2}f$ we can rewrite (3.8) as

$$\begin{aligned} \underset{g \in \mathbf{R}^n}{\text{minimize}} \quad & g'D^{-1/2}LD^{-1/2}g \\ \text{subject to} \quad & g'D^{1/2}\mathbf{1} = 0 \\ & ||g||^2 = vol(V). \end{aligned} \tag{3.9}$$

Now, making the observations that $D^{-1/2}LD^{-1/2} = L_{sym}$, that $D^{1/2}\mathbf{1}$ is the first eigenvector of L_{sym} and that $vol(V)$ is constant, we can identify the problem (3.9) to be on the form of (8.1) and we can apply Theorem 8.2.2, (see Appendix B), and its solution g is given by the second eigenvector of L_{sym}. Substituting back $f = D^{-1/2}\mathbf{g}$ and using Proposition 8.2.1 in Appendix B we can see that f is the second eigenvector of L_{rw}, or equivalently, the generalized eigenvector of $Lu = \lambda Du$. Hence, the solution of the relaxed problem (3.8) is given by the $f = u$. So, we can approximate the minimizer of (3.1) by the second eigenvector of L_{rw}. However, since the eigenvector takes values in $\mathbf{R}^n$ the solution must be discretized to satisfy the constraints on the discrete indicator vector f. In the case when $k = 2$ this is done by using the sign of f as indicator function, that is

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \bar{A} & \text{if } f_i < 0 \end{cases} \tag{3.10}$$

This result can be extended for the case $k > 2$, by instead of f defining the indicator vectors $h_j = (h_{1,j},...,h_{n,j})'$, $(i = 1,..,n$ and $j = 1,...,k)$ by

$$h_{i,j} = \begin{cases} \frac{1}{\sqrt{vol(A_j)}}, & \text{if } v_i \in A_j \\ 0 & \text{otherwise.} \end{cases} \tag{3.11}$$

Next we set the matrix H to be the matrix with the k indicator vectors as its columns, i.e $H = \{h_j\}_{j=1}^{k}$. Now, since $HH' = I$, $h_i' D h_i = 1$, and that $h_i' L h_i = \frac{cut(A_i,\bar{A}_i)}{vol(A_i)}$, the *k-way* $C_N(\Pi)$ minimization problem (3.1) can be reformulated as

$$\begin{aligned} \underset{A_1,\ldots,A_k}{\text{minimize}} \quad & Tr(H'LH) \\ \text{subject to} \quad & H \quad \text{as in} \quad (3.11) \\ & H'DH = I. \end{aligned} \tag{3.12}$$

Again, relaxing the discreteness condition on h_j, and introducing T by $T = D^{1/2}H$, we can write the relaxed problem in the following way

$$\begin{aligned} \underset{T \in \mathbf{R}^{n \times k}}{\text{minimize}} \quad & Tr(T'D^{-1/2}LD^{-1/2}T) \\ \text{subject to} \quad & TT' = I. \end{aligned} \tag{3.13}$$

System (3.13) is a standard trace minimization problem and its solution is obtained by choosing the matrix T to contain the k first eigenvectors of L_{sym} as columns. Again, substituting back $H = D^{-1/2}T$ and using Proposition 8.2.1 in Appendix B, we see that H will consists of the first k eigenvectors of the matrix L_{rw}, or equivalent to the first k generalized eigenvectors of $Lu = \lambda Du$. This results in the normalized spectral algorithm from [9] for arbitrary k.

3.1.2 Maximizing Modularity

Several graph partition formulations with modularity maximization have been proposed. Here we only present the integer formulation introduced in [27]. Other commonly used formulations include the spectral relaxation presented by [28], this has great similarities with the relaxed spectral formulation presented for the normalized cut. The reader is referred to [28] for a comparison between these formulations.

The formulation in [27] results in a linear integer program. The objective is to find a partition Π of V that maximizes the modularity as defined in (2.12). Note that in this formulation the number of clusters k is not fixed.

First we introduce the variable f_{ij} for each pair (i,j) of vertices in the graph, where

$$f_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ belong to the same cluster} \\ 0, & \text{otherwise.} \end{cases} \tag{3.14}$$

These variables can be interpreted as an equivalence relation over V and thus form a partition by its equivalence classes. To ensure consistency we must impose the following constraints on the relation.

$$\begin{array}{ll} \text{reflexivity} & \forall i : f_{ii} = 1 \\ \text{symmetry} & \forall i,j : f_{ij} = f_{ji} \\ \text{transitivity} & \forall i,j,l : f_{ij} + f_{jl} - 2f_{il} \leq 1. \end{array} \tag{3.15}$$

Using the introduced decision variables f_{ij} the objective function (2.12) can be expressed as

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2m} \right] f_{ij} \tag{3.16}$$

where as before m is the total number of edges in the graph and d_i denotes the degree of vertex i.

The modularity maximization problem is then given by

$$\begin{array}{ll} \underset{f_{ij}}{\text{maximize}} & Q \\ \text{subject to} & f_{ij} \text{ as in (3.15)}, \quad f_{ij} \in [0,1]. \end{array} \tag{3.17}$$

Since we consider undirected graphs, we have $f_{ij} = f_{ji}$, so it is enough to introduce $\binom{n}{2} = O(n^2)$ optimization variables f_{ij} for $i < j$. However, there are $\binom{n}{3}$ constraints from (3.15). Brandes et al. showed in [27] several characteristics of modularity maximization, including a proof that the decision version of the problem is NP-complete.

3.2 Algorithms

Considering the complexity of both the formulations in the previous section the algorithms presented here will be heuristic. When solving (3.17) we will use a greedy agglomerative approach, similar to the ones presented in [29, 30, 31], while (3.12) will be solved using the spectral relaxation (3.13) and the approach described in [26].

3.2.1 Spectral Algorithm for Normalized Cut

A partition from minimizing the normalized cut will be found by considering the relaxed problem (3.13). This problem is computed by solving the generalized eigenvalue problem for L. The obtained relaxed solution must then be made feasible for the original problem, taking the discrete constraints into consideration. Several approaches have been proposed for this including directional cosine method, randomized projection heuristic, and clustering rounding. We will adapt the method suggested in [26], using k-means algorithm on the eigenvectors of the normalized Laplacian L_{rw} to obtain a feasible solution.

The complete algorithm can be described by the following steps.

Algorithm 4: Spectral normalized cut

```
Input: Adjacency matrix $A$ $(n \times n)$, number of clusters $k$
Result: Clusters, $(C_1,...,C_k)$
$D = diag(d_1,...,d_n)$ ;
$L = D - A$ ;
Compute the $k$ first generalized eigenvectors $(u_1,...,u_k)$ by solving the generalized
eigenvalue problem $Lu = \lambda Du$ ;
$U = [u_1...u_k]$ ;
$Y = [\ ]$ ;
for $i$ = 1:n do
    $y_i = U(i,:)$ ;
end
Cluster the points $(y_i)_{i=1}^n$ in $\mathbf{R}^k$ by MATLAB kmeans function into clusters
$(C_1,...,C_k)$ ;
```

The complexity of Algorithm 4 is determined by the computation of the k first eigenvectors of $L_{rw} = D^{-1}L$, which in general has complexity $O(n^3)$. However, using sparse matrices this can be done more efficiently using a power method or Krylov subspace methods such as the Lanczos method.

3.2.2 Greedy Algorithm for Modularity

Problem (3.17) is solved by using a greedy agglomerative hierarchical heuristic that follows a scheme similar to [30]. The algorithm is based on an aggregation process with two different phases. The first phase performs small changes by shifting nodes between clusters and a second phase merges entire clusters, resulting in larger changes. Starting from singleton clusters the algorithm evaluates the modularity change in every phase, ΔQ, of each possible move/merging and then performs the action that would result in the largest modularity increase. The algorithm will alternate between these two actions as long as an improvement in the modularity is possible. The algorithm can be described with the following pseudo-code.

```
Algorithm 5: Greedy Modularity
  Input: Adjacency matrix, $A$
  Result: Clusters, $(C_1,...,C_k)$ and $Q$
  Initialize clusters, with one node per cluster;
  while change == true do
      while node move == true do
          Pick a node at random and choose its best move based on $\Delta Q$ choosen
          from (3.18).
      end
      while cluster merge == true do
          Pick a cluster at random and choose its best merging based on $\Delta Q$
          choosen from (3.19).
      end
  end
```

Where ΔQ is the modularity change of each possible node move or cluster merging. Using (2.12) moving a vertex i from its current cluster c_i to another cluster c_j will in the first phase result in the modularity change

$$\Delta Q_{i,c_i,c_j} = \frac{1}{2m}\left(-(\sum_{k\in c_i} A_{ik} - A_{ii}) + 2\cdot\frac{d_i(W_{c_i}-d_i)}{2m} + \sum_{k\in c_j} A_{ik} - 2\cdot\frac{W_{c_j}\cdot d_i}{2m}\right) \tag{3.18}$$

with $W_{c_j} = vol(c_j) = \sum_{k\in c_j} d_k$ introduced to simplify notations. The first term removes $i's$ contribution of internal edges in c_i. The second and fourth term adds and removes the null factor term associated with moving i from c_i to c_j. The third term adds the contribution of i to the internal edges of c_j.

The modularity change from merging cluster c_i and cluster c_j in the second phase is computed using the relation from [29] as

$$\Delta Q_{c_i,c_j} = 2(e_{c_ic_j} - b_{c_i}b_{c_j}) \tag{3.19}$$

where $e_{c_ic_j}$ is the fraction of edges with ends in c_i and c_j and $b_{c_i} = \sum_{c_j} e_{c_ic_j}$ is the fraction of all ends of edges being attached to any of the vertices in cluster c_i.

The algorithm terminates when no moves in both phases can produce a positive ΔQ. The authors of [30] writes that the overall complexity of the algorithm is not straightforward to establish but each of the two phases iterates over the edges of the nodes, resulting in an overall $O(m)$ complexity. This was further supported by the simulations in [30].

4

Empirical study of power law graphs

In this chapter we present some result from an empirical study of different instances of power law graphs from the models discussed in Section 2.3.3. The first motive for this study is that the loose definition of a power law graph enables graphs with very different network characteristics to fit within the definition. Hence, two graphs with similar power law degree distribution can differ vastly in terms of other network metrics. The aim is to highlight common features for all power law graphs as well as differences between the models. Also, the structure of the genetic graphs will be compared to the characteristics of a real-life Market graph instance, created from closing prices on the American Stock market. This is done to evaluate how well the models can represent the topology of a market graph.

4.1 Model generated graphs

This section studies the topological characteristics of the graph models discussed in Section 2.3.3. The models considered are Barabasi-Albert (BA), Power law random graph (PLRG) and the Copying model (COPY). The graphs are created by selecting input parameters such that the desired power law exponent is 2.5 for all the generators except Barabasi-Albert, which can only produce graphs with power law exponent 3. Graphs of sizes between 500 and 5000 vertices were generated and studied.

4.1.1 Degree distribution

Since the main purpose of the models presented is to generate graphs with a degree distribution approximating a power law the degree sequences of the generated graphs are studied and the power law exponent is approximated by the maximum likelihood method from [32] for validation purpose. The graphs below show the degree distribution of the generated graphs plotted together with an approximated power law for graphs with 3000 vertices. Table 4.1 reports the estimated power law exponents of the probability density function obtained from ML-estimation of 20 generated graphs with 3000 vertices and the variance between the different estimations.

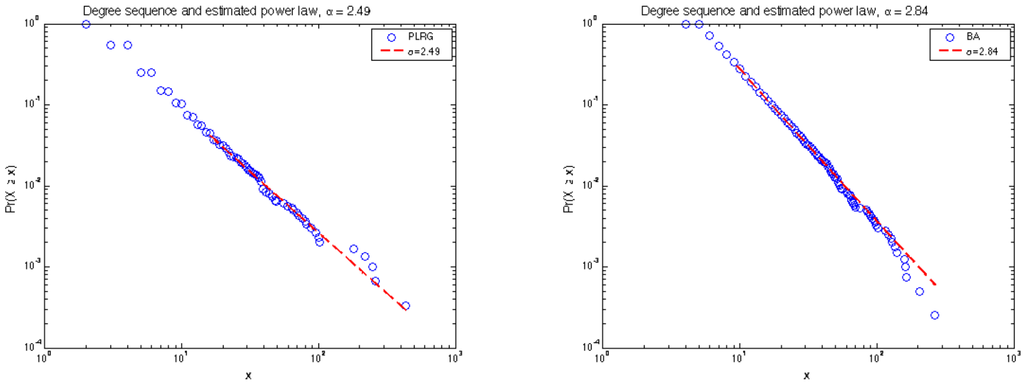
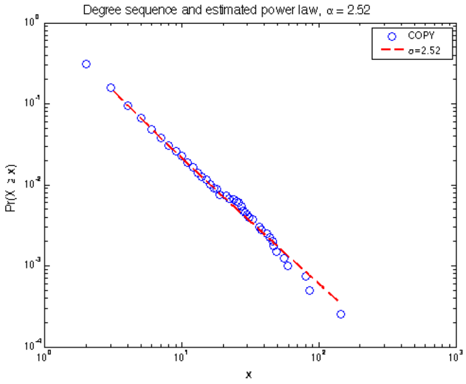


Figure 4.1: Degree distribution and approximated power law for PLRG, BA, COPY graphs.



4.1.2 Clustering coefficient

Another characteristic of many power law graphs is a high clustering coefficient, this is examined for the different graph models. For comparison, the edge density, $\delta(G)$ of

Power exponent	μ	σ^2
PLRG	2.558	0.0118
Barabasi	2.920	0.0105
Copy	2.482	0.0979

Table 4.1: Power law exponent for generated graphs, $N = 3000$.

the graphs are also included. Table 4.2 reports the mean and variance of the global graph clustering coefficient together with the edge density of the graph. One can see that the global clustering coefficient for all the generated power law graphs are higher compared to their edge density. However, the clustering coefficient of the COPY graphs are much higher (relative the graph edge density) than for graphs generated by the BA model.

Graph clustering coefficient	μ	σ^2	$\delta(G)$
PLRG	0.1045	$1.31 \cdot 10^{-3}$	0.0018
Barabasi	0.0141	$4.47 \cdot 10^{-6}$	0.0020
Copy	0.0165	$1.67 \cdot 10^{-4}$	$8.02 \cdot 10^{-4}$

Table 4.2: Mean and variance of global graph clustering coefficient, edge density, $N = 3000$.

4.1.3 Assortativity

The assortativity coefficient R measures the correlation between the node degrees in the network. A positive R indicates an assortative network, meaning that high degree nodes are linked to other high degree nodes. A negative R suggests dissortative behaviour in the network, where high degree nodes are connected to low degree nodes, creating hubs in the network. The definition we use was introduced by Newman in [33] as

$$R = \frac{1}{\sigma_q^2} \sum_{jk} jk(e_{jk} - q_k q_j). \tag{4.1}$$

Where q_k is the distribution of the remaining degree of the vertices, reflecting the number of edges encountered when reaching a vertex by traversing an edge. This is given by $q_k = \frac{(k+1)p_{k+1}}{\sum_j p_j}$, with p_j being the probability of a random node having degree j. The link distribution e_{jk} is the joint probability distribution of the remaining degrees of the two vertices at either end of a randomly chosen edge. In other words, the probability that a vertex with remaining degree k is connected to a vertex with remaining degree j. Also, σ_q denotes the standard deviation of the distribution q_k. For undirected network we have that $e_{jk} = e_{kj}$ and $\sum_{jk} e_{jk} = 1$. Newman showed [33] that in practice for an observed network R is computed from

$$R = \frac{m^{-1}\sum_i j_i k_i - [m^{-1}\sum_i \frac{1}{2}(j_i + k_i)]^2}{m^{-1}\sum_i \frac{1}{2}(j_i^2 + k_i^2) - [m^{-1}\sum_i \frac{1}{2}(j_i + k_i)]^2} \tag{4.2}$$

where j_i and k_i are the degree at each end (vertex) of the edge $i = 1,...,m$.

Table 4.3 show estimated assortativity coefficient computed from 20 instances of each graph model and market graph instances with $\theta = [0.2 : 0.1 : 0.7]$.

Assortativity	μ	σ^2
PLRG	-0.0796	$8.81 \cdot 10^{-4}$
Barabasi	-0.03	$9.07 \cdot 10^{-4}$
Copy	-0.1206	$1.31 \cdot 10^{-4}$
Market graph	-0.1028	$3.6 \cdot 10^{-3}$

Table 4.3: Assortativity coefficient for generated graphs, $N = 3000$ and market graphs with $\theta = [0.2 : 0.1 : 0.7]$

4.1.4 Shortest path

Another network topology measure is the mean shortest path of the graph (also called average path length). Mathematically this can be expressed as $l_(G) = \frac{1}{n(n-1} \cdot \sum_{i \neq j} v(i,j)$, where $v(i,j)$ is the length of the shortest path between node i and j. This metric reflects how fast information is spread in the network. Previous result indicates that this is smaller for many power law graphs than for uniform random graphs [17]. Using graphs of size 500, and implementing the algorithm by *Dijkstra*, [34] the mean shortest path was found for graphs of size 500 (mean over 20 model generated graphs) was found. Results are reported in Table 4.4.

Shortest Path	μ	σ^2	max
PLRG	3.55	0.22	7
Barabasi	3.22	0.094	5
Copy	4.86	0.54	10
Market graph 0.6	3.5266	0.6548	11
Market graph 0.7	5.4450	1.1333	14

Table 4.4: Mean, variance and max of shortest path in generated graphs. $N = 500$. and market graphs with $\theta = 0.6, 0.7$.

4.2 The Market graph

By considering the closing prices of stocks on the New York Stock market (comprising of NYSE, Nasdaq, AMEX) a market graph was created. The original data consisted of 504 observations of 6330 stocks taken from *Yahoo Finance* with observations made between January 4:th 2012 and December 31 2013.

In order to obtain more reliable results two pre-processing procedures were applied on the original data. First, all illiquid instruments were removed. This was done by removing all instruments that had no trading volume for more than 20% of the observations. The second filtering procedure was introduced due to the large amount of Exchange traded funds, (ETF's) present on the American market. The ETF's were removed since they often aim to track the market itself making them highly correlated with most stocks in the market. Their presence adds a noise of highly correlated instruments, not reflecting the overall behavior of the market. After applying these two procedures 4519 instruments remained, these time series were used to construct the market graph and its adjacency matrix A_{ij} by using equations (2.35) and (2.36) in Section 2.4.

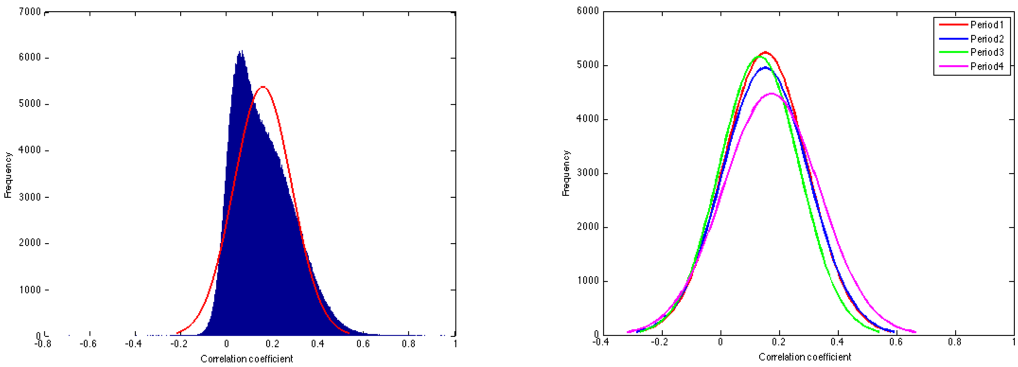


Figure 4.2: a) Correlation distribution and fitted distribution for entire period, b) Fitted correlation distribution for different time periods.

4.2.1 Correlation distribution

The correlation distribution represents the fundamental structure of the market. A plot of the correlation distribution for the entire time period can be found in the left hand graph in Figure 4.2 together with a fitted normal distribution, with $\mu = 0.1532$ and $\sigma = 0.1264$. One can see that the correlation distribution of the US market does not seem to fit perfect with the normal distribution. Even though both tails of the distribution are covered the shape of the fitted curve is not consistent with the data. However, it is interesting to note that stocks seem to mainly exhibit positive correlation, suggesting that stock prices will often move in the same direction. This has been observed before and has then been interpreted as a sign of globalisation with the motivation that more and more stock effect each other positively, [2, 35]. The graph on the right in Figure 4.2 shows fitted distributions for different, shorter time periods, each period consisting of 100 observations. Even though there are some differences between the different periods the correlation distribution of the market remains stable over the considered time intervals.

4.2.2 Edge density

The density of the market graph will of course depend on the correlation threshold. Varying the threshold θ generates graphs of different degrees of correlation. Figure 4.3 shows the edge density for different threshold, as expected the density will decrease with increasing threshold.

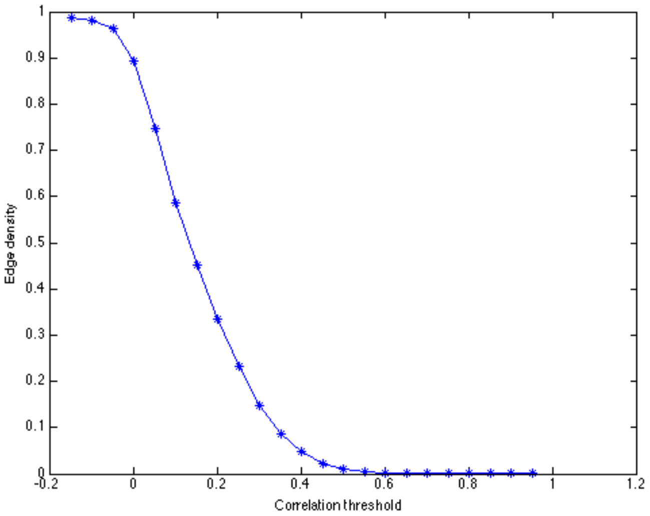


Figure 4.3: Edge density as a function of correlation threshold

4.2.3 Clustering coefficient

By computing the global clustering coefficient for graphs of different θ we found that the cluster coefficient was larger among positively correlated stocks than for negatively correlated stocks. As an example, the edge density of the graph obtained with threshold 0.6 is very close to that of the complementary graph for threshold -0.05. However, the corresponding global clustering coefficients of the two graphs are $C = 0.76$ and $C = 0.19$ respectively. Hence, one can suspect that positively correlated stocks tend to cluster more in the graph than negatively correlated stocks. This feature has been observed previously for other market graphs [1, 2].

For higher positive thresholds the global clustering coefficient appears almost constant. Figure 4.4 shows the graphs clustering coefficient for positive θ. For all $\theta \in [0.2, 0.9]$ the clustering coefficient of the graph remains in the interval $[0.70, 0.82]$. It should be noted that this is significantly higher than what would be expected from an uniform random graph with the same edge density. A high clustering coefficient is a common feature among many real life graphs, indicating that the market graph could possess a community structure.

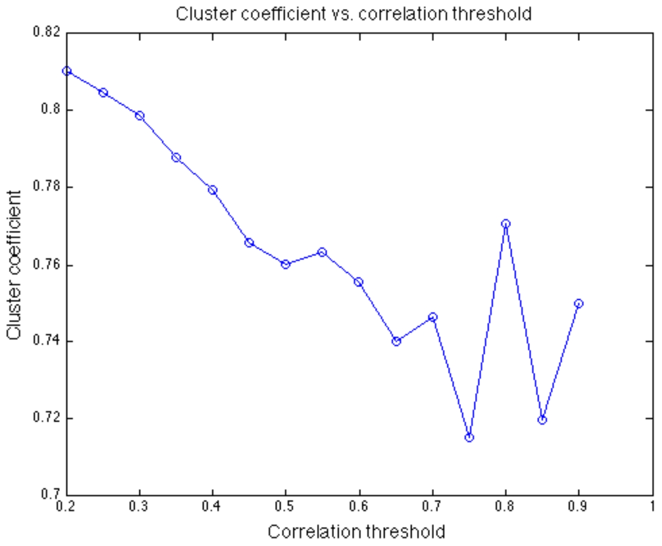


Figure 4.4: Cluster coefficient as a function of correlation threshold

4.2.4 Assortativity

Figure 4.5 displays the average degree of neighbours plotted against the node degree for the market graph with threshold $\theta = 0.5$. The graph shows that the market graph does not possess any clear assortative or dissortative behavior. This seem to be the case especially for the low degree nodes, where the spread of the neighbours degree is the greatest. However, for nodes with higher degrees the behaviour seems slightly dissortative, indicated by the negative slope for higher degrees.

Also, by computing the assortativity coefficient 4.1 for different $\theta \in [0.2, 0.7]$ we find that $R \in [-0.2, -0.05]$ for all these values, indicating a weak dissortative behavior.

4.2.5 Connected components

By studying the connectivity of the graph for a series of thresholds it was found that the graph becomes disconnected for a threshold above $\theta = 0.12$. It therefore becomes interesting to study how the connected components of the graph changes depending on the threshold. Figure 4.6 shows the size of the connected components in the graph plotted against positive thresholds θ. The left graph of Figure 4.6 shows the size of the largest connected component while the right hand sub-figure shows the sizes of the second to fifth largest connected components against the correlation threshold θ.

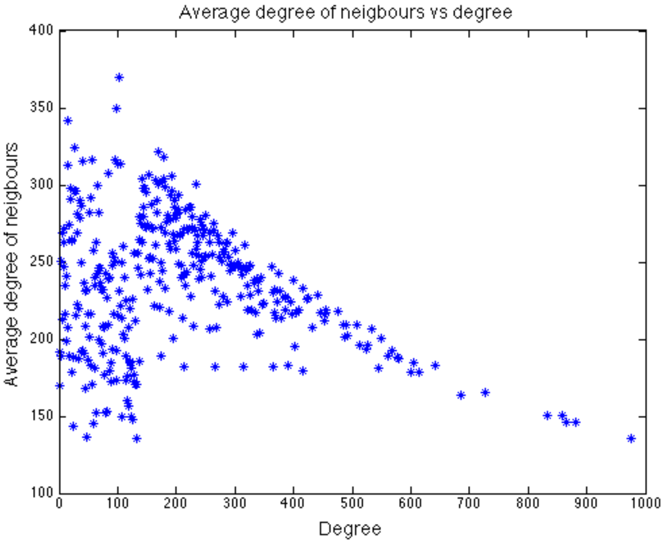


Figure 4.5: Average degree of neighbour against node degree for $\theta = 0.5$

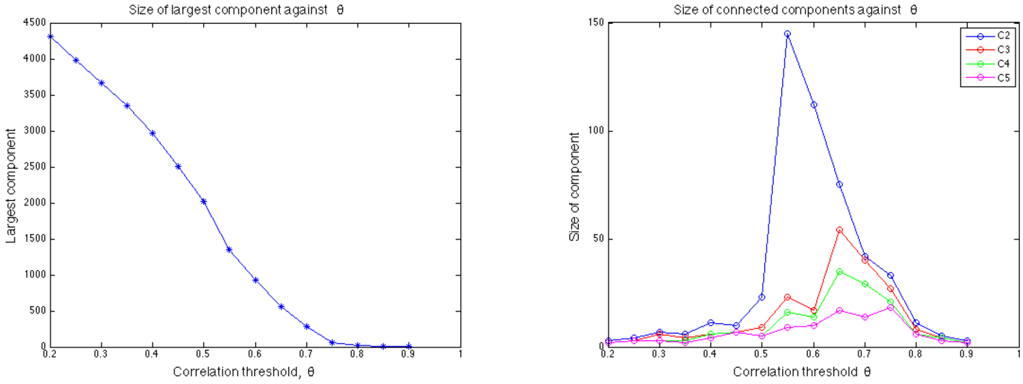


Figure 4.6: a) Size of largest connected component, b) Size of connected components 2-5.

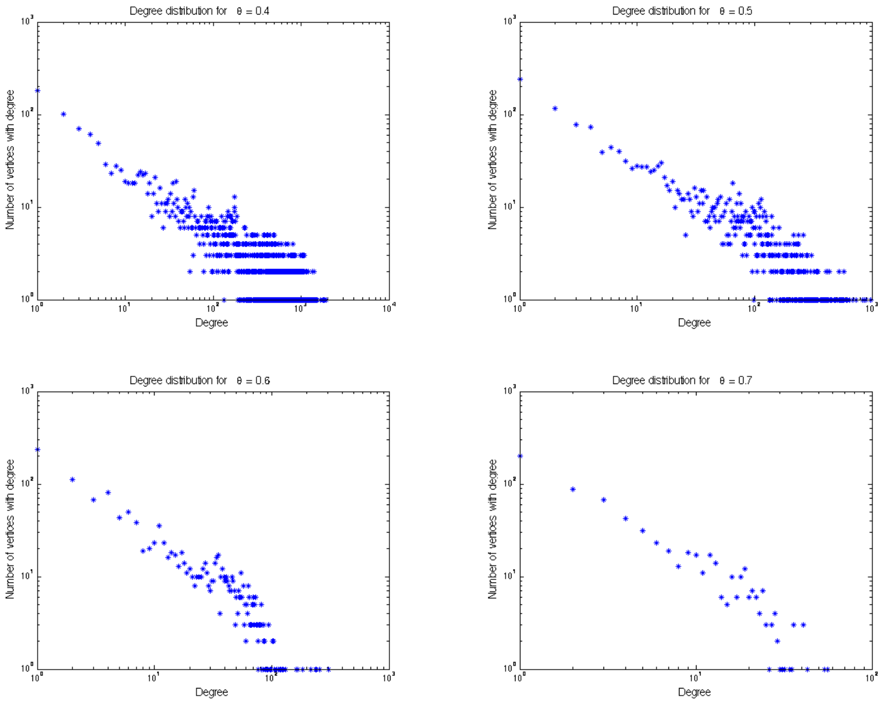


Figure 4.7: Degree distribution for $\theta = \{0.4, 0.5, 0.6, 0.7\}$

4.2.6 Degree distribution

By fixing the threshold a specific market graph is obtained. For this graph a degree distribution can be studied. As was also found in [2] the degree distribution is filled with noise for lower thresholds, however for higher values the power law behaviour becomes more clear. Figure 4.7 shows the degree distribution for thresholds $\theta = \{0.4, 0.5, 0.6, 0.7\}$ in a loglog plot. From the figure one can notice that the noise in the graph decreases as the threshold is increased. Also, it is interesting to note that the slope is lower compared to the edge distribution of many other real life graphs. For instance, the Web graph has been estimated to follow a power law with slope 2.18 [14]. The small exponent suggests that there could exist many vertices with high degree in the graph implying that there could exist larger clusters in the graph.

4.3 Graph models for representing the Market graph

Taking only the studied metrics into consideration we can make the following observations about the topologies of the simulated graphs relative the topology of the market graph. First, all of the studied models does indeed create graphs with a power law degree distribution. However, since the model by Barabasi-Albert only produce power law exponents equal to 3 this is inappropriate for modelling the market graph. The model also fails to capture both the high clustering coefficient of the market graph and its slightly dissortative behaviour.

The PLRG model can generate power laws with varying exponent, however it produces graphs with low clustering coefficient relative the market graph. The model is also a bad representation for a dynamic market since the it creates a graph in one single step, not allowing the network size to grow over time. Also, the model requires that we know the explicit degree distribution of the network, something that is not always possible in practice.

Finally, the COPY model is the only model that somewhat captures the high clustering coefficient of the market graph. Also, it seems to produce weakly dissortative networks, similar to the market graph. However, the COPY model was more difficult to tune and showed larger deviations in many metrics.

5

Result from simulations

5.1 Simulation Setup

By applying Algorithm 4 and 5 on instances of model generated graphs and on market graph instances partitions were obtained. In a first attempt to evaluate the result of the two approaches we ran both algorithm N times on each studied graph. For each of these N partitions the following metrics were computed.

- **Number of clusters found** - Remember that this is a free variable for the Modularity based formulation while it is fixed for the Normalized cut approach.
- **Internal clustering density** - computed from Equation (2.4)
- **Max internal cluster density** - computed from Equation (2.3)
- **Min internal cluster density** - computed from Equation (2.3)
- **External cluster density** - computed from Equation (2.5)
- **Min cluster size**
- **Max cluster size**

Then, the average over all N values were taken, and the results are reported in Table 7.1 and 7.2 in Appendix A. Since the formulation based on Normalized cut requires a fixed number of clusters, k as input, this algorithm was applied with three different $k = 10, 20, 30$ for each graph.

Additionally, to evaluate the consistency of the partitions for each algorithm we compute the Adjusted Rand Index (ARI) [36] between some obtained partitions. The ARI measures the overlapp between two partitions and is defined in the following way. Given a set $V = (1,2,...,n)$ and the partitions $X = (X_1,..., X_s)$ and $Y = (Y_1,...,Y_t)$ of V we can define the following quantities

a - the number of pairs of elements in V that are in the same set in X and in the same set in Y.

b - the number of pairs of elements in V that are in different sets in X and in different sets in Y.

c - the number of pairs of elements in V that are in the same set in X and in different sets in Y.

d - the number of pairs of elements in V that are in different sets in X and in the same set in Y.

Using these quantities the Adjusted Rand index is defined as

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]}. \tag{5.1}$$

ARI has expected value 0 and maximal value 1, corresponding to identical partitions. It can be used to compare both partitions obtained by the same approach, to evaluate the method's consistency as well as comparing the results obtained from different formulations. Computed ARI for different pairs of partitions can be found in Table 7.3 in Appendix A.

The considered graphs consists of Market graph instances created by the different thresholds $\theta = \{0.4, 0.5, 0.6, 0.7\}$, and genetic graphs from the models PLRG, BA and COPY. Specific graph characteristics are reported in Table 7.1 and 7.2 in Appendix A.

5.2 Partitions of the Market graph

The result from the simulations show that for the lower thresholds $\theta \in [0.4, 0.5]$, both approaches produce partitions with low modularity. Also, the partitions have low minimal internal cluster density and a high external density relative to the overall graph density. As an example, for $\theta = 0.4$ the minimal internal density of a cluster is of the same magnitude as the edge density of the entire graph for both algorithms. Also, the external cluster edge density is approximately half that of the overall graph density, indicating that the identified clusters are not well separated. All these results indicate that the Market graph lacks a strong community structure for lower thresholds. This is not really surprising as at lower thresholds even weakly correlated stocks can be connected in the graph making it more difficult to distinguish which instruments truly form clusters.

Also, for these lower thresholds the size of the largest cluster found is very large, especially for the greedy modularity approach where the largest cluster consists of nearly 2/3

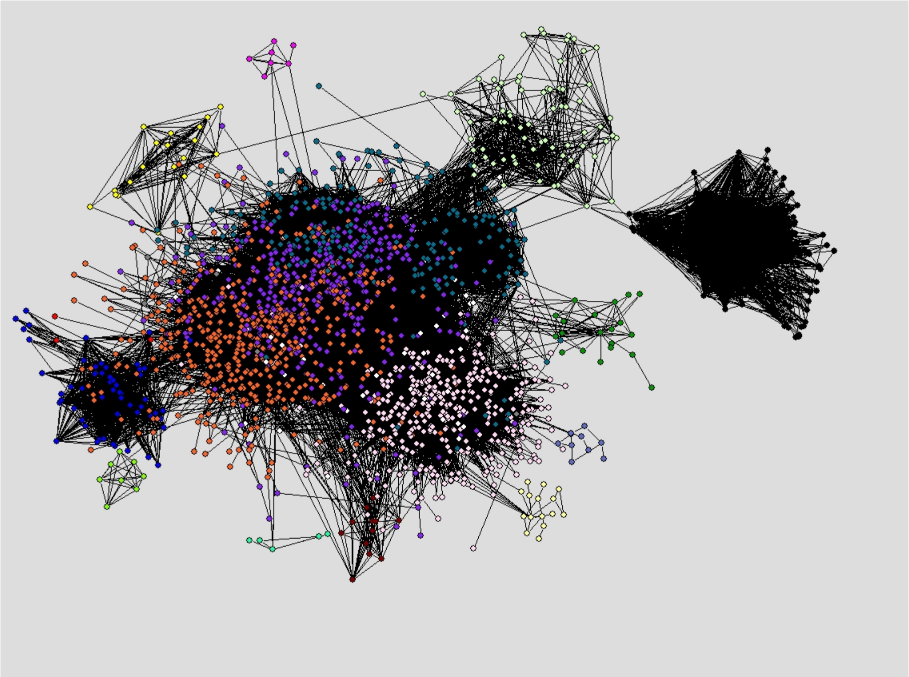


Figure 5.1: Partition of th Market graph ($\theta = 0.5$) into 18 clusters from Modularity approach

of the considered nodes. This cluster has low internal density and is strongly connected to the rest of the graph. This further supports the idea that the market graph lacks a clear community structure for lower thresholds. Figure 5.1 shows a partition of the giant connected component obtained by the modularity approach for $\theta = 0.5$. One can see that even though each cluster seems strongly connected, most of them are not well separated.

For higher values of the thresholds ($\theta \in [0.6, 0.7]$) the quality of the partitions increases. Partitions of these graphs display higher modularity, combined with higher minimal internal cluster density and lower external density. As an example, for $\theta = 0.7$ the minimal internal cluster density is more than three times as high as the overall graph density and the external cluster density is less than $\frac{1}{10}$ the edge density of the entire graph. Hence, clusters are both more dense and better separated compared to partitions for lower thresholds. This result was found for all the approaches. Figure 5.2 and 5.3 show the partitions of the largest connected component for $\theta = 0.7$ for both algorithms. In this case the partitions seems very similar, this is also confirmed by computing the Adjusted Rand index for the two partitions, $ARI = 0.9295$, further indicating a large overlap between the two partitions.

In Figure 5.4 we have plotted the internal cluster density against cluster size for both approaches (with 20 partitions of each) applied on Market graphs with $\theta = [0.5, 0.6, 0.7]$. From Figure 5.4 we can notice that the result from the two approaches becomes more

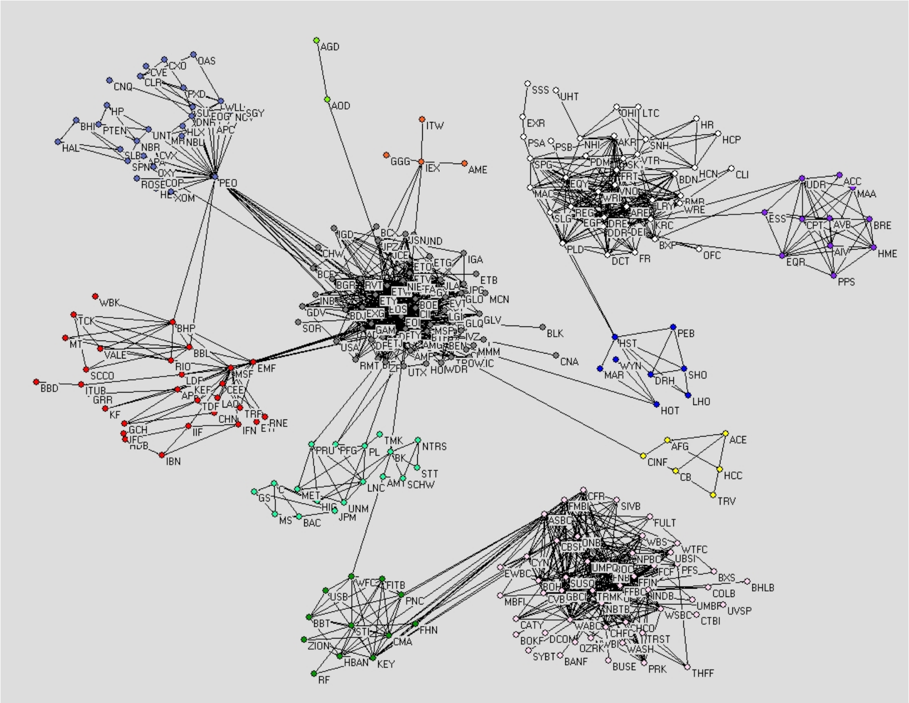


Figure 5.2: Partition from modularity approach for Market graph $\theta = 0.7$

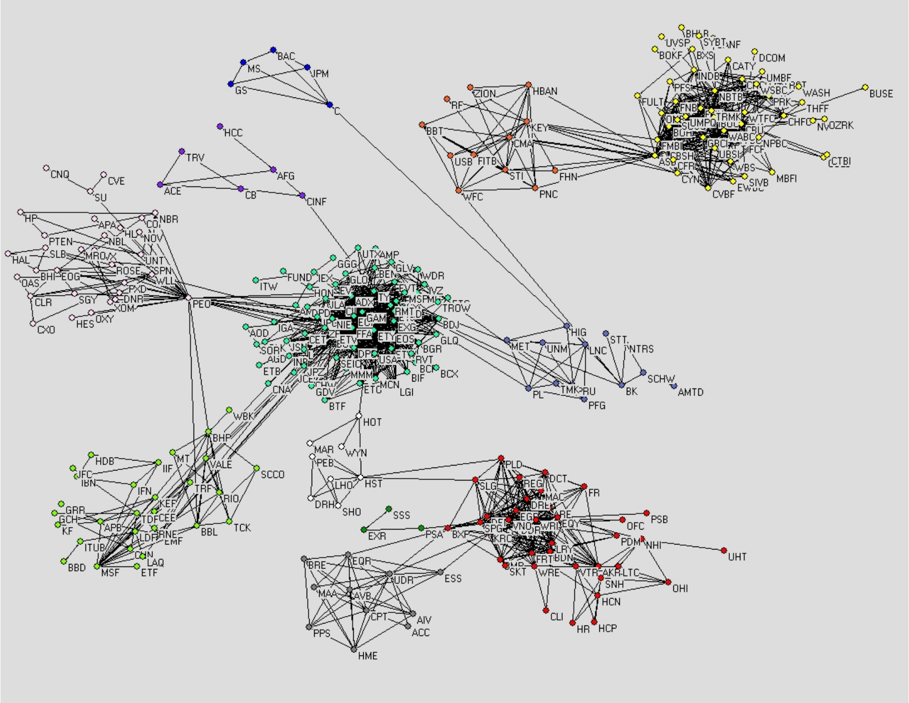


Figure 5.3: Partition from normalized cut approach for market graph $\theta = 0.7$

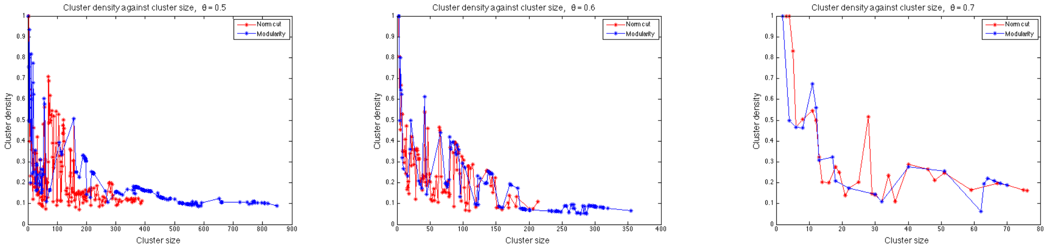


Figure 5.4: Internal cluster density against cluster size for GM (blue) and SC (red) $\theta = [0.5, 0.6, 0.7]$

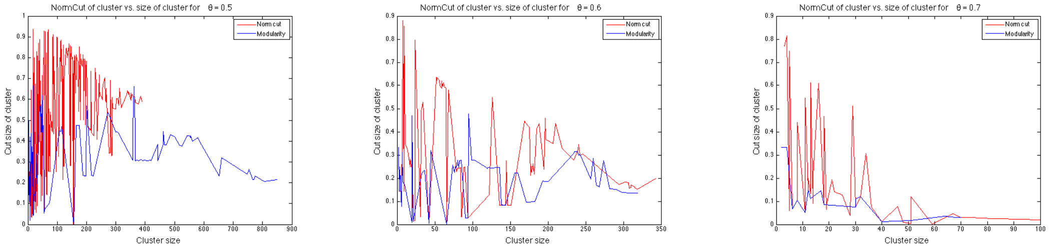


Figure 5.5: Normilized cut of cluster vs. cluster size for GM (blue) and SC (red) $\theta = [0.5, 0.6, 0.7]$

similar for larger values of θ, indicating a stronger community structure for higher thresholds in the graph. Figure 5.5 shows the normalized cut plotted against cluster size of the corresponding partitions. Here we can notice a difference between the approaches since the modularity formulation produces partitions with smaller fluctuations in the normalized cut of clusters than the cut formulation.

Comparing the two approaches one can notice that using modularity generally gives a larger giant cluster that the method using normalized cut (when they are set to find the same number of clusters).

5.2.1 Internal cluster structure

Since a common way to classify instruments in portfolio management is by dividing them into industrial sectors we will compare the internal structure of the clusters with 12 industrial sectors of the market. First, the sector representation in the data and for the giant component of different market graphs can be found in Table 7.4 in Appendix A. It is especially interesting to note that as the threshold θ increases some sectors as finance, basic industries and energy increases its percentage in the largest connected component of the graph while other sectors, as health, drastically decreases. Hence, the degree of correlation between industrial sectors in the market differs.

To analyse the internal structure of the clusters in the graph partitions we study the percentage representation of each sector in every cluster. Table 7.5 in Appendix A show the relation between industrial classification and clusters in partitions of the giant connected component of the market graph, obtained by the greedy modularity approach for different thresholds.

For lower thresholds $\theta = 0.5$ the sector overlap between clusters is relatively large. This is especially clear for the two largest clusters in the partition. In these sets all 12 industry sectors are represented. Hence, for moderate correlations there is no strong connection between clusters and industry sectors. It is also interesting to note that the financial sector stands out by existing in all of the 4 largest clusters, indicating that this sector is connected to many other sectors at this correlation level. This is further confirmed by the fact that when selecting the 10 nodes with the highest degrees in the graph more than 50% of these belong to the financial sector.

For higher thresholds the correlation between industrial sectors and clusters is stronger. At threshold 0.7, no cluster includes stocks from all sectors and more clusters now only consists of one sector. However, even though the correlation between cluster and sector is very strong, it is not complete, even at this high threshold level. This phenomena introduces the possibility to use graph clusters instead of industry sectors in portfolio diversification.

5.2.2 Dynamics of partition

To study possible dynamics and stability in the partition structure we divide our data of price returns into 4 periods, each consisting of 150 days, and with 50 days overlap between each consecutive period. For each time series we construct a Market graph for $\theta = 0.7$.

First, computing the giant connected component (GC) of each market graph we notice that its size varies greatly, from 253 instruments in period 3 up to 701 in period 4. Thus the market correlations in these sub-periods are quite different compared to the correlations obtained by using data covering all time series. Moreover, by considering the overlap for these different GC's we can see that it is not only the cardinality that changes but also which instruments that are present in the GC. The intersection between all the GC's is 131 indicating that the instruments composing the GC's changes over different time periods. However, the edge density of the GC is almost constant over all 4 periods.

Using the greedy modularity approach, partitions for the different time period were obtained (20 for each graph), the results are reported in Table 7.5. From these numbers it can be observed that the modularity decreases between period 1 and period 4. This, together with the fact that the external cluster density increases over the same time could imply that the the community structure of the graph decreases over time. Partitions

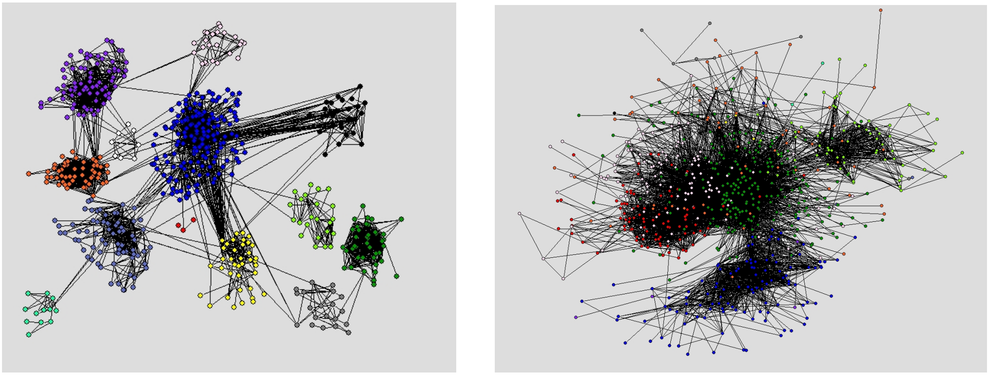


Figure 5.6: Partitions of market graph with $\theta = 0.7$ for period 1 (left) and 4 (right) into 11 resp. 12 clusters.

of the market graph for $\theta = 0.7$ from period 1 and 4 can be found in Figure 5.6, these graphs confirm the differences between the partitions.

By studying the sector composition of the clusters it was found that the connection between sectors and clusters was weaker in all sub periods compared to the entire period. Also, this pattern increased for every considered period, and in the last period most clusters consisted of several different industrial sectors. Hence, the correlation between industrial sectors and graph clusters is weaker for shorter time series.

5.3 Partitions of Model generated graphs

The two approaches were also applied on some instances of the model graphs from Section 2.3.3. Since the graphs are random, 5 instances each with 1000 nodes and power law exponent, β close to 2.5 (except for Barabasi-Albert graph where $\beta = 3$) were generated. The procedure described in section 5.1 was performed, only with the exception that for each graph the algorithms were applied $N = 10$ times. The results are reported in Table 7.2.

Figures 5.7, 5.9, 5.8 show examples of the partitions obtained from the greedy modularity algorithm for each type of graph. From Figure 5.7 it is clear that the Barabasi-Albert graphs lacks a community structure. This is supported by the numbers reported in Table 7.2. Partitions of BA graphs have relatively low modularity and a high external cluster density. Also, the internal cluster density is almost constant regardless of the cluster size, indicating that it does not exist an optimal size for the clusters in the partition.

A partition of a PLRG graph can be seen in Figure 5.8. The quality of this partition seems better compared to the one for the BA graph, and the community structure is more clear. However the clusters are still quite connected to each other, supported by the relatively large external cluster density of the partition.

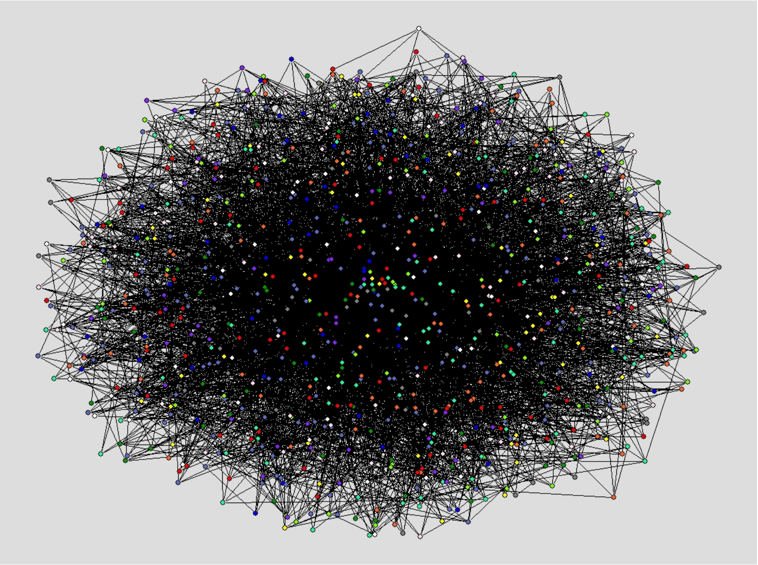


Figure 5.7: Partition from modularity approach for Barabasi Albert

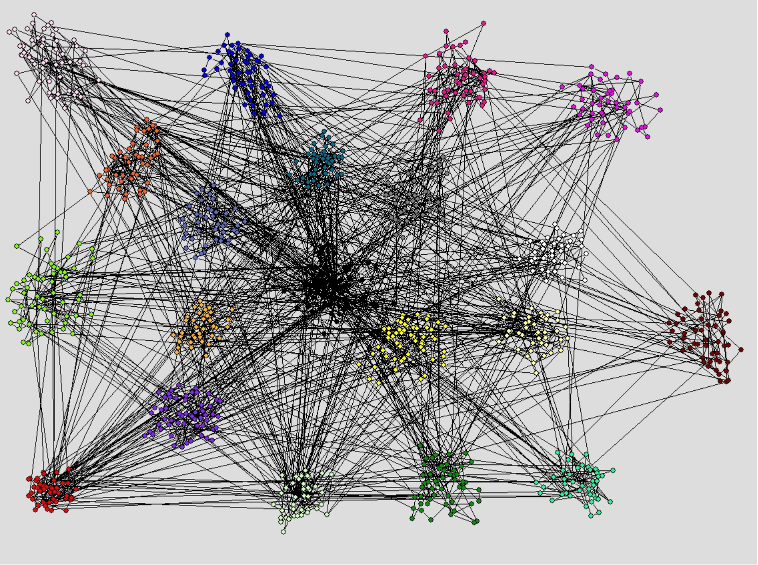


Figure 5.8: Partition from modularity approach for PLRG graph

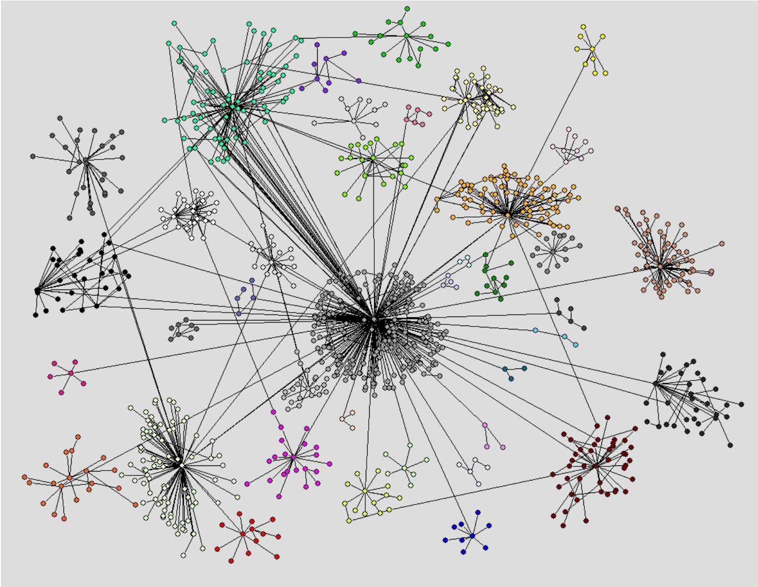


Figure 5.9: Partition from modularity approach for Copy graph

Finally, Figure 5.9 shows the partition of a COPY model graph. For this graph we can clearly distinguish a community structure in the graph. The obtained clusters in the partition are well separated and clusters have an internal density much higher than that of the entire graph.

Another indication that neither BA nor PLRG graphs exhibit a clear community structure can be seen from Figure 5.10 showing the modularity of a partition (found by the Spectral algorithm for different k) plotted against the number of clusters. First, the modularity of the COPY graph is almost twice as large as for the BA and PLRG graphs. Also, the maximum modularity for this graph is not constant for all number of clusters, which indicates that the topology has an optimal cluster partition.

To summarize, from the simulations we can make the following general observation

- Barabasi-Albert graphs seems to lack a clear community structure. Partitions of these graphs show low modularity, combined with low internal cluster density and high external cluster density relative the global graph density. Moreover, these result are constant no matter the number of clusters in the partition. This indicates that the topology lacks such an optimal partition structure.
- PLRG graphs exhibit some community structure, but due to large deviation in the results we can not conclusively claim it has clear community structure.
- COPY model exhibit strong community structure.

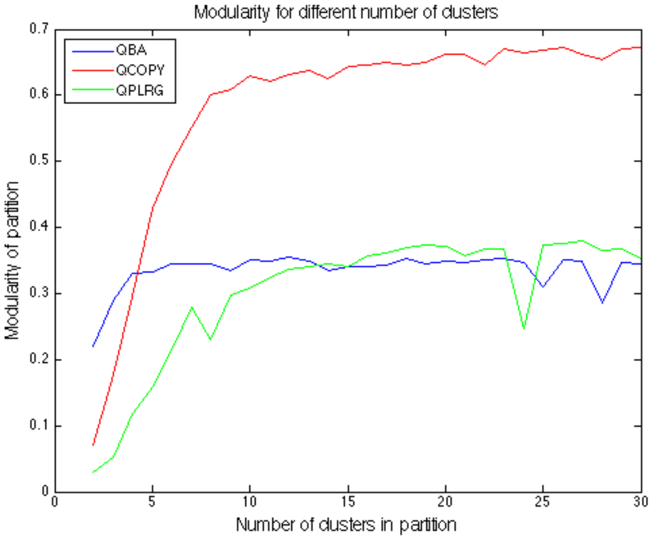


Figure 5.10: Modularity of of partition plotted against the number of clusters in partition for different power law graphs.

6

Conclusions

This chapter concludes the findings of the thesis. We also discuss possible modifications and recommendations for future research.

We have, by implementing two different formulations for graph partitioning studied the partition structure of the Market graph, a network constructed by daily price returns on the American stock market. The two approaches are based on two commonly used metrics to identify graph clusters, the normalized cut and modularity. Depending on which objective that was used to identify the clusters we found partitions with different structure. One major difference between the results from the two approaches was that the modularity based formulation produced partitions with a large spread between the clusters sizes while for the spectral approach the clusters are more equal in size. We also found that the partitions from each formulation became more similar for market graphs with a higher correlation threshold.

Results from both formulations indicated that the Market graph lacks a strong community structure for correlation thresholds below 0.6. However, for thresholds above this the partition, the community structure of the graph becomes more clear. Thus, we can note that even though features as high clustering coefficient and a power law degree distribution can be observed in the market graph for thresholds lower than 0.6, the community structure is not present at those correlation levels.

The fact that not all power law graphs possess a clear clustered structure was further supported but the results from partitions of the considered generated graphs. The structure and quality of the partitions differ vastly among the different power law models. Hence, we find that there is no direct connection between the occurrence of a power law degree distribution and that the graph has a community structure. The great topological differences between the graph models was also seen from the empirical study of their topology. The findings in Chapter 3 showed that none of the considered graph models is able to fully capture the structure of the Market graph. However, this should come as no surprise as the considered models were developed to mimic the topology of the Internet graph rather than the Market graph. These observations illustrate the need to develop models specifically for the topology of the Market graph.

By studying the internal structure of clusters we showed that the connection between clusters and industrial sectors increased with the partition quality (modularity, or cut size). This result supports that, when considering longer time periods, prices of instruments within a particular industrial sector are often significantly correlated.

Considering partitions for different, shorter sub-periods showed that the partition structure and its quality fluctuates over time. We also observed that the connection between clusters and industrial sectors increased with the length of the considered time period and was stronger in the early sub-periods in the data.

Possible focus for future studies could be finding stricter formulations for the cluster densities in the partition. Also, the connection between graph clusters and industrial sectors could be further studied to examine the possibility of using clusters as instrument classification in portfolio analysis. Another topic could be to consider weighted graphs. Finally, it would be interesting to further study partitions for more and longer time series to analyse the dynamics in the partition structure.

7

Appendix A: Tables

7.1 Partitions of the Market graphs

Number of runs is $N = 50$ for all Market graphs

	Threshold θ			
Market graph	0.4	0.5	0.6	0.7
Nr. nodes in GC	2964	2020	934	278
Global density (GC)	0.0928	0.0400	0.0282	0.0337
Greedy Modularity				
Mean nr. clusters	14.1	20.66	13.66	11.08
Max/Min nr. clusters	[9,20]	[15 33]	[10 19]	[10 12]
Mean internal density	0.5768	0.4686	0.4562	0.3758
Max internal density	0.9968	0.9571	0.9933	0.8691
Min internal density	0.0924	0.0996	0.0714	0.1009
External cluster density	0.0400	0.0123	0.0037	0.0013
Min cluster size	2.1	2.5	2.64	3.32
Max cluster size	938.78	630.22	276.78	66.48
Modularity	0.2406	0.4353	0.6738	0.7271
Spectral Ncut, $k = 10$				
Nr. clusters	10	10	10	10
Mean internal density	0.3243	0.2765	0.3162	0.3514
Max internal density	0.7932	0.6511	0.6506	0.8282
Min internal density	0.1083	0.0526	0.0632	0.117
External cluster density	0.0689	0.0188	0.0051	0.0021
Min cluster size	13.40	11.70	9.40	5.15
Max cluster size	1086	976	280.8	71.45
Modularity	0.0490	0.1360	0.6290	0.7044
Spectral Ncut, $k = 20$				
Nr. clusters	20	20	20	20
Mean internal density	0.2865	0.3037	0.3596	0.5113
Max internal density	0.8373	0.7370	0.8168	0.9950

Min internal density	0.0925	0.0874	0.0745	0.1524
External cluster density	0.0852	0.024	0.0079	0.0043
Min cluster size	7.25	8.80	4.10	2.60
Max cluster size	484.35	351.0	157.85	54.95
Modularity	0.0417	0.2844	0.5984	0.6349
Spectral Ncut, $k = 30$				
Nr. clusters	30	30	30	30
Mean internal density	0.2576	0.3084	0.4412	0.5526
Max internal density	0.8150	0.7967	0.9837	1
Min internal density	0.0801	0.0979	0.0880	0.1599
External cluster density	0.0901	0.0274	0.0094	0.0108
Min cluster size	9.62	4.5	2.6	2
Max cluster size	257.75	227.5	127.4	43.7
Modularity	0.0489	0.2559	0.5654	0.5575

Table 7.1: Result for the algorithms on market graphs for different θ

7.2 Partitions of Model generated graphs

Number of runs is $N = 10$ for all model generated graphs.

	Power law graph model		
Graph	CO	PLRG	BA
Nr. nodes in GC	1000	1000	1000
Global density (GC)	0.0020	0.0055	0.0137
Greedy Modularity			
Mean nr. clusters	38.4	16.2	13.66
Max/Min nr. clusters	[37,40]	[14 18]	[15 21]
Mean internal density	0.2359	0.0476	0.0454
Max internal density	0.6667	0.1316	0.0766
Min internal density	0.0192	0.0254	0.0341
External cluster density	0.00086	0.00300	0.0017

Min cluster size	3	23.9	28.85
Max cluster size	219.1	94.35	73.75
Modularity	0.8228	0.4039	0.3522
Spectral Ncut, $k = 10$			
Nr. clusters	10	10	10
Mean internal density	0.0325	0.0610	0.0293
Max internal density	0.0766	0.1684	0.0495
Min internal density	0.0076	0.0081	0.0194
External cluster density	0.000162	0.0017	0.0033
Min cluster size	31	14.6	59.7
Max cluster size	346.1	447.5	152.3
Modularity	0.6359	0.2933	0.3281
Spectral Ncut, $k = 20$			
Nr. clusters	20	20	20
Mean internal density	0.067	0.0664	0.0513
Max internal density	0.1757	0.2494	0.0788
Min internal density	0.0188	0.0214	0.0351
External cluster density	0.000584	0.0024	0.0035
Min cluster size	13.5	9.7	31.6
Max cluster size	120.4	140.1	73.3
Modularity	0.714	0.3879	0.3565
Spectral Ncut, $k = 30$			
Nr. clusters	30	30	30
Mean internal density	0.0901	0.0858	0.0785
Max internal density	0.3074	0.2834	0.1232
Min internal density	0.0248	0.0329	0.0372
External cluster density	0.000603	0.0025	0.0036
Min cluster size	7.3	7.4	18.7
Max cluster size	94.9	77.6	87.7
Modularity	0.7233	0.4062	0.3500

Table 7.2: Result for algorithms on generated graphs

7.3 Adjusted Rand Index

Adjusted Rand Index for different partition approaches and market graphs. The ARI was obtained by choosing 3 pairs of partitions, at random from each type, computing the ARI for each of them and then taking the average of these values. (GM - Greedy modularity, SNC - Spectral Normilized cut).

	Threshold θ		
Market graph	0.5	0.6	0.7
ARI			
GM & GM	0.5846	0.7845	0.9593
SNC & SNC	0.5317	0.9124	0.9212
GM & SNC	0.4084	0.5900	0.8963

Table 7.3: Adjusted Rand Index for different algorithms and Market graphs

7.4 Industrial Sectors in the Market graph

Tables 7.4 shows the percentage of different sectors in the data and market graphs for different thresholds.

		Threshold θ		
Market graph	Data	0.5	0.6	0.7
Sector				
Basic industries	0.0755	0.1065	0.1192	0.0294
Capital goods	0.0860	0.1097	0.1205	0.0196
Consumer Dur.	0.0326	0.0368	0.0314	0.0049
Consumer Non Dur.	0.0508	0.0387	0.0226	0
Consumer Serv.	0.1525	0.1503	0.1267	0.2745
Energy	0.0712	0.0958	0.1192	0.1569
Finance	0.1649	0.2245	0.2961	0.5049
Health	0.1128	0.0323	0.0138	0.0049
Miscellaneous	0.0315	0.0146	0.0013	0
Public Util.	0.0590	0.0729	0.0803	0
Technology	0.1379	0.0977	0.0590	0.0049
Transportation	0.0252	0.0203	0.0100	0

Table 7.4: Sector representation in data and market graphs GC for $\theta = [0.5, 0.6, 0.7]$.

7.5 Industrial sectors and clusters

Cluster	Size	Bas.Ind	Cap.Go	Cons.D	Cons.N	Cons.S	Ener	Fin	Heal	Missc.	Pub.U	Tech	Trans
1	188	0.01	0	0	0	0.61	0.01	0.03	0	0	0.35	0	0
2	29	0.07	0	0.07	0.59	0	0	0	0.21	0	0.07	0	0
3	157	0	1	0	0	0	0	0	0	0	0	0	0
4	758	0.12	0.16	0.06	0.04	0.09	0.18	0.09	0.04	0.02	0.02	0.16	0.02
5	3	0	0	0	0	0	0	0	0	1	0	0	0
6	5	0	0	0	0	1	0	0	0	0	0	0	0
7	3	0	0	0	0	0.67	0	0	0	0.33	0	0	0
8	10	0	0	0	0	1	0	0	0	0	0	0	0
9	15	0	0	0	0	0	0	0	0	0	0	1	0
10	57	1	0	0	0	0	0	0	0	0	0	0	0
11	8	0	0	0	0	0	1	0	0	0	0	0	0
12	20	0	0	0	0	0.1	0.4	0	0	0	0.5	0	0
13	2	0	0	0	0	0	0	1	0	0	0	0	0
14	407	0.04	0.08	0.03	0.03	0.05	0.01	0.66	0.03	0.02	0.02	0.06	0.01
15	52	0.12	0.37	0.08	0.02	0.08	0	0	0.02	0	0	0	0.31
16	7	1	0	0	0	0	0	0	0	0	0	0	0
17	32	0	0	0	0	0.94	0	0.06	0	0	0	0	0
18	3	0	0	0	0	0.33	0	0	0	0	0.67	0	0
19	262	0.06	0.08	0	0.04	0.04	0.06	0.33	0.04	0.02	0.27	0.04	0.04
20	2	0	0	0	0	0	1	0	0	0	0	0	0

Market graph $\theta = 0.5$ with industrial sectors and clusters from Greedy modularity.

Cluster	Size	Bas.Ind	Cap.Go	Cons.D	Cons.N	Cons.S	Ener	Fin	Heal	Missc.	Pub.U	Tech	Trans
1	173	0.01	0.02	0.01	0.01	0	0	0.94	0.01	0	0	0.01	0.01
2	76	0.01	0	0	0	0.03	0.95	0	0	0	0.01	0	0
3	66	0.02	0	0	0.03	0	0.02	0	0	0	0.94	0	0
4	33	0	0.15	0	0.09	0	0	0	0	0	0	0.76	0
5	94	0	0	0	0	0.98	0	0.02	0	0	0	0	0
6	21	0.1	0.8	0.05	0.05	0	0	0	0	0	0	0	0
7	7	0	0	0	0	0	0	0	0	0	0	0	1
8	7	0	0.86	0	0	0	0	0	0	0	0	0.14	0
9	42	1	0	0	0	0	0	0	0	0	0	0	0
10	3	0.33	0	0.67	0	0	0	0	0	0	0	0	0
11	123	0.27	0.08	0.02	0.03	0.03	0.2	0.31	0.02	0	0.01	0.02	0
12	265	0.16	0.31	0.11	0.05	0.06	0.03	0.14	0.05	0.01	0	0.1	0
13	4	0	1	0	0	0	0	0	0	0	0	0	0
14	20	0	0	0	0	0	0	1	0	0	0	0	0

Market graph $\theta = 0.6$ with industrial sectors and clusters from Greedy modularity.

Cluster	**Size**	**Bas.Ind**	**Cap.Go**	**Cons.D**	**Cons.N**	**Cons.S**	**Ener**	**Fin**	**Heal**	**Missc.**	**Pub.U**	**Tech**	**Trans**
1	6	0	0	0	0	0	0	1	0	0	0	0	0
2	2	0	0	1	0	0	0	0	0	0	0	0	0
3	32	0	0	0	0	0	1	0	0	0	0	0	0
4	11	0	0	0	0	1	0	0	0	0	0	0	0
5	4	0	0.5	0.25	0	0	0	0	0	0	0	0.25	0
6	40	0	0	0	0	1	0	0	0	0	0	0	0
7	51	0	0	0	0	0	0	1	0	0	0	0	0
8	30	0.5	0	0	0	0	0.08	0.42	0	0	0	0	0
9	64	0	0.14	0	0	0	0	0.79	0.07	0	0	0	0
10	30	0	0	0	0	0	0	1	0	0	0	0	0
11	8	0	0	0	0	1	0	0	0	0	0	0	0

Market graph $\theta = 0.7$ with industrial sectors and clusters from Greedy modularity.

7.6 Partitions of Market graph for different periods

Number of runs is $N = 20$.

		Threshold θ		
Market graph	P1	P2	P3	P3
Nr. nodes in GC	532	559	253	701
Global density (GC)	0.0313	0.0209	0.0273	0.0291
Greedy Modularity				
Mean nr. clusters	9.1	12.5	10	13.45
Max/Min nr. clusters	[7, 11]	[11 13]	[8 12]	[10 16]
Mean internal density	0.2853	0.2614	0.3281	0.4988
Max internal density	0.7047	0.5613	1	1
Min internal density	0.0643	0.0511	0.09264	0.0727
External cluster density	0.0012	0.0012	0.0025	0.0070
Min cluster size	8.3000	8.1000	2.0	2.0
Max cluster size	134.4	120.5	62.8	239.7
Modularity	0.7483	0.7847	0.6143	0.5120

Table 7.5: Partitions for different time periods and $\theta = 0.7$ from Greedy modularity approach

8

Appendix B: Graph Laplacians

8.1 The unnormalized Laplacian

Definition The unnormalized graph Laplacian matrix of a graph G with adjacency matrix $A = [a_{ij}]_1^n$ and degree matrix $D = diag(d_1,...d_n)$ is defined $L = D - A$.

Proposition 8.1.1 *(Properties of L.)*
L satisfies the following properties:

1. *For every vector $f \in R^n$ it holds that*

$$f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(f_i - f_j)^2$$

2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is zero, corresponding to the eigenvector of the constant one vector, **1**.*
4. *L has n non-negative, real-valued eigenvalues, $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.*

8.2 The normalized Laplacians

Definition We define two different normalized Laplacians L_{rw} and L_{sym} as,

$$L_{rw} = D^{-1}L = I - D^{-1}W$$

$$L_{sym} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2}$$

Proposition 8.2.1 *(Properties of L_{sym} and L_{rw})*
The normalized Laplacians L_{sym} and L_{rw} satisfy the following properties.

1. *For every vector $f \in R^n$ it holds that*

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

2. *λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2}u$*

3. *λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generelized eigenproblem $Lu = \lambda Du$*

4. *0 is an eigenvalue of L_{rw} with the constant one vector as eigenvector, $\mathbf{1}$. 0 is an eigenvector of L_{sym} with eigenvector $D^{1/2}\mathbf{1}$*

5. *L_{rw} and L_{sym} are positive semi-definite and have n non-negative, real-valued eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.*

Consider the symmetric eigenvalue problem

$$A\mathbf{x} = \lambda M\mathbf{x}, \quad A = A^*, \quad M = M^* > 0 \tag{8.1}$$

Theorem 8.2.2 (Trace theorem for the generalized eigenvalue problem)
Let A and M be as in (8.1), then

$$\lambda_1 + \lambda_2 + ... + \lambda_p = mintrace(X^* AX)$$

where λ_1,...,λ_p are the eigenvalues of (8.1). Equality holds if and only if the columns of the matrix X that achieves the minimum span also the eigenspace corresponding to the smallest p eigenvalues.

Bibliography

[1] V.Boginski, S.Butenko, P.M.Pardalos, Mining Market Data: A Network approach, Computers and Operations Research 61 (2006) 23171–3184.

[2] V.Boginski, S.Butenko, P.M.Pardalos, On Structural Properties of the Market Graph, Innovations in financial and economic networks. (2003) 29–45.

[3] A.Koldanov, B.Goldengorin, A.Vizgunov, V.Kalyagin, P. Pardalos, Network approach for the russian stock market, Computational Management 11 (2013) 44–55.

[4] J.Pattillio, N. Youssef, S.Butenko, Clique relaxation models in social network analysis, in: M.T.Thai, P.M.Pardalos (Eds.), Handbook of Optimization in Complex Networks, Springer New York, 2012, pp. 143–162.

[5] S. E. Schaeffer, Survey: Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.

[6] Y. Cai, Y. Sun, Esprit-tree: hierarchical clustering analysis of millions of 16s rrna pyrosequences in quasilinear computational time, Nucleic Acids Research-Method papers.

[7] S. Fortunato, Community detection in graphs, Physical Review E 486 (2010) 75–174.

[8] H. Matsuda, T. Ishihara, A. Hashimoto, Classifying molecular sequences using a linkage graph with their pairwise similarities, Theoretical Computer Science 210 (2) (1999) 305 – 325.

[9] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[10] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2) (2004) 1–15.

[11] A. P.Erdös, The Evolution of Random Graphs, Magyar Tud. Akad. Mat. Kutató Int. Közl. 5 (1960) 17–61.

[12] B.Bollobas, Random Graphs, 2nd Edition, Cambridge Studies in Advanced Mathematics, Cambrige University Press, 2001.

[13] R. Diestel, Graph Theory, 4th Edition, Vol. 173 of Graduate texts in mathematics, Springer, 2012.

[14] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology, SIGCOMM Comput. Commun. Rev. 29 (4) (1999) 251–262.

[15] D. T.Bu, On Distinguishing between Internet Power Law Topology Generators, Proceedings-IEEE INFOCOM (2002) 638–647.

[16] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (2002) 47–97.

[17] M.Hernandez, T.Kleiberg, H.Wang, P. Mieghem, A qualitative comparison of power law generators, International Symposium on Performance Evaluation of Computer and Telecommunication System 11 (2007) 17–30.

[18] W. Aiello, F. Chung, L. Lu, A random graph model for massive graphs, in: Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC '00, ACM, New York, NY, USA, 2000, pp. 171–180.

[19] W.Aiello, F.Chung, L.Lu, A Random Graph Model for Power Law Graphs, Experimental mathematics 10:1, (2001) 53–66.

[20] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[21] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, A. S. Tomkins, The web as a graph: Measurements, models, and methods, in: Proceedings of the 5th Annual International Conference on Computing and Combinatorics, COCOON'99, Springer-Verlag, Berlin, Heidelberg, 1999, pp. 1–17.

[22] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal, Stochastic models for the web graph, in: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS '00, IEEE Computer Society, Washington, DC, USA, 2000, pp. 57–.

[23] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: Densification laws, shrinking diameters and possible explanations, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05, ACM, New York, NY, USA, 2005, pp. 177–187.

[24] W.-Q. Huang, X.-T. Zhuang, S. Yao, Network analysis of the Chinese stock market, Physica A 338 (2009) 2956–2964.

[25] V.Boginski, S.Butenko, P.M.Pardalos, Statistical analysis of financial networks, Computational Statistics and Data analysis 48 (2005) 431–443.

[26] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[27] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, D. Wagner, On modularity clustering, IEEE Transactions on Knowledge and Data Engineering 20 (2) (2008) 172–188.

[28] M. E. J. Newman, Spectral methods for community detection and graph partitioning, Physical Review E 88 (4) (2013) 42822.

[29] M. E. Newman, M. Girvan, Fast algorithm for detecting community structure in networks, Physical Review E 69 (2) (2004) 18–33.

[30] E. L. Martelot, C. Hankin, Fast multi-scale detection of relevant communities, Comput. J. 1136–1150.

[31] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment (2008) 1–12.

[32] A. Clauset, C. R. Shalizi, M. E. J. Newman, Power-law distributions in empirical data, SIAM Rev. 51 (4) (2009) 661–703.

[33] M. E. Newman, Assortative Mixing in Networks, Physical Review Letters 89 (20) (2002) 208701.

[34] E. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1) (1959) 269–271.

[35] V.Boginski, S.Butenko, P.M.Pardalos, Modelling and Optimization in Massive Graphs, American mathematical society 48 (2003) 17–39.

[36] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1) (1985) 193–218.

TRITA-MAT-E 2014:54
ISRN-KTH/MAT/E—14/54-SE

www.kth.se