IDENTIFYING AFFINITY GROUPS OF RESEARCHERS IN FEP THROUGH THE APPLICATION OF COMMUNITY DETECTION ALGORITHMS

by

André Martinez Candeias Lima

Thesis for the Master's Degree in Modelling, Data Analysis and Decision Support Systems

Supervised by

Prof. João Gama
Prof. Pavel Brazdil

2017

Biographical note

André Martinez Candeias Lima was born in Porto in September 1992. After graduating from Escola Secundária Filipa de Vilhena, he enrolled in 2010 in the School of Economics and Management of University of Porto, where he would conclude his Bachelor's Degree in Economics in July 2013. Two years later he enrolled once again in the same Faculty, this time in the Master's Degree in Modelling, Data Analysis and Decision Support Systems. From September 2016 to March 2017 he also worked as a Functional Analyst at OSI, a company that belongs to Grupo Amorim.

Acknowledgements

I would like to thank everyone who contributed to this project. First and foremost, my supervisors Professor João Gama and particularly Professor Pavel Brazdil for his unending support, wisdom and enthusiasm to drive this project forward.

I would also like to thank Rui Sarmento and Luís Trigo, who provided the tool Affinity Miner and who helped me along the way. I would like to thank the team at Authenticus for providing the data and the members of the Faculty who showed their interest in this project.

I would like to thank everyone at OSI for the support given in the months I worked there.

I would like to thank my friends and family for their support during this journey. A special thanks goes out to my friends from OT:C, who gave me many moments of distraction and fun.

I am specially grateful to my parents, my sister Jessica and my friends Filipa, Maria João, Daniel, Mike and Sara, who were always available to hear me out and always kept me on the right path to finish this thesis.

Resumo

Redes são utilizadas para representar agentes e as suas interações. Como centro de investigação, a Faculdade de Economia da Universidade do Porto pode beneficiar ao ser representada como uma rede. Com recurso ao Affinity Miner podemos criar uma rede onde cada nó é um investigador, e estes estão ligados de acordo com o grau de similaridade entre os títulos das suas publicações. Podemos descobrir comunidades nesta rede que nos dão novas perspetivas relativas à sua estrutura.

Um dos objetivos deste estudo é descobrir uma boa representação da rede original. Esta tem um número elevado de ligações, e para a simplificar são removidas ligações com base num *limiar* abaixo do qual as ligações são considerads fracas. Isto afeta os resultados obtidos com a aplicação dos algoritmos de deteção de comunidades, logo o limiar e o algoritmo devem ser analisados conjuntamente. Indicadores comuns, como a *modularidade*, não se revelaram adequados para esta tarefa. Criei uma medida, *modularidade com penalização de componentes*, que me permite determinar um intervalo para o limiar no qual se obtêm bons resultados para os algoritmos. Analiso depois os algoritmos neste intervalo e concluo que o *método de Louvain* alcança a melhor performance.

Determinar se existe uma estrutura pertinente na rede é também importante. Analiso uma rede simplificada e descubro que os investigadores mais centrais pertencem ao agrupamento científico de *Economia*. O *método de Louvain* obtém 9 comunidades, que estão na sua maioria preenchidas por investigadores que pertencem ao mesmo grupo científico. Na minha opinião e na de alguns membros da Faculdade, a estrutura descoberta é pertinente e permite obter novas perspetivas em relação à organização da Faculdade.

Abstract

Networks are used to represent agents and their interactions. As an investigation centre, the School of Economics and Management of the University of Porto can benefit from being represented as a network. Using Affinity Miner we can create a network where each node is a researcher, and they are connected by the level of similarity of the titles of their publications. Communities can be discovered in this network, giving us insight into its structure.

One of the objectives of this study is to discover a good representation of the original network. It has a high number of links, and to simplify it links are removed based on a *threshold* under which links are considered weak. This affects the results obtained when applying community detection algorithms, so both the threshold and algorithms have to be analysed together. Common indicators, such as *modularity*, prove to be inadequate for this task. I have created a *modularity with component penalty* measure which allows me to determine an interval of thresholds that provide good results from the algorithms. I then test the algorithms for this particular interval and conclude that *Louvain's method* achieves the best performance.

Determining if a pertinent structure is present in the network is also important. I analyse a simplified network and discover that the most central researchers are from the scientific group of *Economics*. I then apply *Louvain's method* and obtain 9 communities. Each of these tends to be filled with researchers from the same scientific groups. Based on my opinion and that of some members of the Faculty, the structure discovered is pertinent and provides interesting insights into the Faculty's organisation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and objectives

Many fields of study resort to networks to represent the underlying entities they are analysing. This representation is useful as long as the entities interact or are related to each other in some way. This allows us not only to better visualise the set of entities but also to uncover certain aspects of the underlying structure. Companies and institutions can also take advantage of this network analysis to better understand their own structure.

Trigo et al. (2015) have created a prototype software called *Affinity Miner*. Its aim is to create networks of researchers based on the articles they have published. It extracts the terms that appear in the titles of the articles published by each author. It constructs a network of researchers linked by the similarity of those terms so the user is able to visually explore the data. Furthermore, the application finds communities in that network which join similar researchers into homogeneous groups, which are called *affinity groups*.

It was recommended to me to examine the underlying community detection algorithms in detail and in particular analyse its behaviour when it is applied to real data. In the first phase I have used the dataset that includes 104 researchers of InescTec. The groups generated by the community detection algorithm implemented in Affinity Miner did not seem well balanced. There were several groups composed of 3 or less researchers,

which seemed rather strange. We have therefore decided to analyse various community detection algorithms in order to achieve better results.

Due to my connection to the School of Economics and Management of the University of Porto (FEP), where I have received my Bachelor in Economics and where I am enrolled in the Master in Modelling, Data Analysis and Decision Support Systems programme, I have decided to carry out our study of community detection algorithms on the data relative to the researchers from FEP. This could prove useful for the Faculty as an investigation centre, aiding in its understand of what actually happens there. It can also be useful for the researchers themselves, helping them search for other colleagues working in similar or complementary fields of study.

1.2 Contributions

This study contributes to the better understanding of the organisation present in FEP from the perspective of published research. I experiment with three community detection algorithms – Walktrap, Louvain and Infomap – with the aim to determine which one obtains the best results. The quality of the results was judged with recourse to two measures called *modularity* and *modularity density*. However, as it turned out later when I was working on the task of simplification of the network, these measures were not the best ones.

One of the contributions is related to the network simplification. The original network has many links, making it difficult to visualise. The simplification method implemented in the Affinity Miner prototype deletes the weakest links. There are not many ways to determine how many links to cut. The level at which this simplification is terminated is determined by a given *threshold*. For this task, I first attempted to use a known measure of the quality of community detection algorithms, known as *modularity*, but it lead to rather strange results discussed later. As such, I had to adapt this indicator and created a *modularity with component penalty* measure to account for some of the flaws of modularity. This new measure was helpful in determining the best thresholds for this network

and also for selecting the best community detection method.

1.3 Organisation of the thesis

This work is organised in 5 chapters. The first chapter has already been presented. In Chapter 2 I review the literature that is relevant for carrying out this study, namely *network analysis*, *community detection* and *information retrieval*.

In Chapter 3 I use the data of FEP researchers to carry out an experimental study of the network simplification and community detection algorithms. They are analysed together since they heavily influence one another. I first analyse the effects of simplifying the network and discover a way to determine the best thresholds. This is then followed by a study of some community detection algorithms applied to the best thresholds, in order to determine which performs better.

In Chapter 4 I discuss the results obtained with a modified version of Affinity Miner that incorporates the best community detection method discussed in the previous chapter. This system is applied to the data of FEP and I carry out an analysis of the network and affinity groups that were discovered. I look at keywords generated by Affinity Miner to summarise each affinity group and compare these groups with the scientific groups defined by the Faculty to determine if the structure found is satisfactory.

Finally, Chapter 5 presents the conclusions of this study and possible future research lines.

Chapter 2

Literature Review

2.1 Network Analysis

Networks can be found everywhere. They are present in society, biology, computer science, and many other fields. It is therefore important to be able to model these networks and extract relevant information from them. This section follows the survey by Oliveira and Gama (2012).

2.1.1 Basic notions of graphs

Networks are usually represented as graphs. Figure 2.1 contains an example of a graph, which is constituted by two basic elements: *nodes* and *links*, which are connections between two nodes. In Figure 2.1, the nodes are circles and the links are lines connecting them. There is different terminology for these depending on the field of study. In mathematics, these are called *vertices* and *edges*. In computer science, these are called *nodes* and *links*/*connections*. In sociology, they are called *actors*/*agents* and *relational ties*. Other areas may also call them *dots* and *arcs*. From the perspective of social networks, it may be easier to interpret the meaning of these nodes and links by referring to them as *individuals* and *connections*. The nodes represent individuals in the network, while

the links represent the relationships between these individuals. In this study I use the terminology of *nodes* and *links*/*connections*/*edges*.

Two additional definitions are those of *neighbourhood* and *path*. The neighbours of a node are the nodes that are directly connected to it. A path is a sequence of nodes in which consecutive pairs of nodes are connected by links.

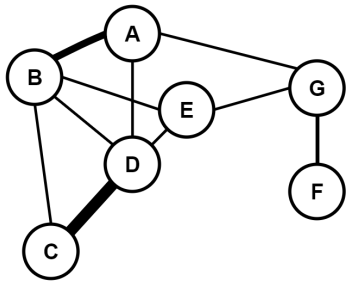

Figure 2.1: Example of a weighted network.

An important distinction is that between *weighted* and *unweighted* graphs. For an unweighted graph, the value of each link is binary, taking the value of 1 when the link between two nodes exists and the value of 0 when it doesn't. A graph can be weighted when the links between the nodes can assume different values, with that value being the weight of the link. Graphically, it can be represented by the thickness of the link, with a thicker line representing heavier weights, as exemplified in Figure 2.1.

It is also important to differentiate between *undirected* and *directed* graphs. In undirected graphs, two nodes are simply connected and there is no direction in the relationship. The order of the nodes does not matter, we can say that node A is connected to node B or that node B is connected to node A. A good example of this is *Facebook*, where people can only be friends, and so saying A is friends with B is equivalent to saying B is friends with A. In directed graphs (or *digraphs*) there is a direction in the relationship between two nodes, and so the order does matter. Graphically, these links are portrayed as arrows indicating the direction of the relationship. A typical example of this is *Twitter*, where users can follow and be followed by different groups of people, and following someone does not mean the user will be followed by that person. As such, saying A is a follower

of B is different than saying B is a follower of A.

Another important concept is that of *connected component*. This is a maximal connected subgraph, which means that there is always at least one path connecting any two nodes in that subset of the graph. On top of that, this subgraph is not connected to the rest of the graph, and thus stands on its own (Easley and Kleinberg, 2010).

With this concept in mind, we can further define a *bridge* as the link between two nodes that would otherwise belong to separate connected components in a graph. The nodes that create these bridges are of particular interest from the social networks perspective, as they become intermediaries between the two groups, allowing for the information to spread from one group to the other. These bridges are very rare in real-world scenarios, since there is usually more than one intermediary between different groups. Granovetter (1973) then defines *local bridges*, which are the links joining two nodes that have no neighbours in common. If we remove this link, the distance between those nodes will increase, since the shortest path between the two would have to go through at least two more nodes, as they do not have neighbours in common. These local bridges distinguish themselves from bridges because they are not necessarily the only available path between the two nodes.

A graph can be represented as an *adjacency matrix*. This is a matrix in which the entries indicate if two nodes are adjacent or not (Wasserman and Faust, 1994). It is a square matrix of i rows and i columns, with a row and a column for each node. The nodes must be in the same order in both rows and columns. The entry x_{ij} indicates if the node in the row i is adjacent to the node in the column j. In an unweighted graph, if the nodes i and j are adjacent then the entry x_{ij} has a value of 1; otherwise it is 0. In a weighted graph, if the nodes are adjacent the entry has the value of the weight of the connection, and a value of 0 otherwise. For undirected graphs, this network is symmetric, with $x_{ij} == x_{ji}$. The elements in the diagonal (x_{ii}) can be given a value of 1 or 0 if we consider self-loops, or they can be *undefined* if we do not consider them, in which case they are represented by a "–". Table 2.1 contains the adjacency matrix of the network in Figure 2.1.

Node	A	B	C	D	E	F	G
A	–	0.7	0	0.5	0	0	0.5
B	0.7	–	0.5	0.5	0.5	0	0
C	0	0.5	–	1	0	0	0
D	0.5	0.5	1	–	0.5	0	0
E	0	0.5	0	0.5	–	0	0.5
F	0	0	0	0	0	–	0.5
G	0.5	0	0	0	0.5	0.5	–

Table 2.1: Adjacency matrix of the example weighted network from Figure 2.1.

2.1.2 Statistical measures for the individuals

The position of a node in the network is measured by its centrality, which attempts to determine the importance of a node in the network by measuring how well connected it is. There are several metrics that are used to compute this. The most popular ones were suggested by Freeman (1978): *degree*, *betweenness* and *closeness*. There have been some attempts at generalising these metrics for weighted networks (Opsahl et al., 2010; Newman, 2001). Bonacich (1987) later introduced *eigenvector centrality*. Another popular metric was proposed by Watts and Strogatz (1998), called *local clustering coefficient*. Some of these are very simple and only consider the number of links a node has, while others go further to consider not just the number of links a node has but also the importance of the nodes it is connected to.

Degree

The *degree* or *valency* of a node is simply the number of neighbours it has, that is, the number of nodes it is directly connected to, or even the number of links it has. For weighted networks the degree can also be called *strength*, and according to Barrat et al. (2004) it is computed by adding the weights of the links of a given node. This is a simple measure, as it is only able to tell us how many direct connections a certain node has regardless of how valuable they are, and as such it does not take the global structure of the network into account. The distribution of the degree in a network usually follows one of a few particular distributions. These are discussed in Section 2.1.4.

Betweenness

Betweenness can be calculated for both nodes and links. *Node betweenness* measures how often the node is between other nodes in the network (Freeman, 1978). The betweenness can be calculated as the fraction of shortest paths between two nodes that pass through the node under analysis, out of all the shortest paths passing through the original two nodes. It is presented in Equation 2.1, where I consider v as the node being measured, b_i as node betweenness of node i, σ_{st} as the number of shortest paths between nodes s and t. $\sigma_{st}(v)$ is the number of shortest paths between the nodes s and t that go through node i. $N(G)$ is the set of nodes in graph G.

$$b_i = \sum_{s,t \in N(G) \setminus i} \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{2.1}$$

For links, this metric it is called *edge betweenness* (Newman and Girvan, 2004) and it measures how often the shortest paths between nodes pass through that link. This helps detecting the bridges and local bridges, since those links tend to have high betweenness. In Equation 2.2 b_e is the edge betweenness of the link e, σ_{st} is the number of shortest paths between nodes s and t, and $\sigma_{st}(e)$ is the number of shortest paths passing through link e.

$$b_e = \sum_{s,t \in N(G)} \frac{\sigma_{st}(e)}{\sigma_{st}} \tag{2.2}$$

Closeness

Closeness (Freeman, 1978) measures how close a node is to the rest of the network, giving an idea about how long it takes to reach the other nodes from there. It is basically the mean length of all the shortest paths from the node under analysis to all the other nodes. Nodes in smaller components of the network will be farther away from the nodes in the larger components. Since there are more nodes in the larger components, the mean length of the shortest paths will be higher for those in the smaller components than it will be for the

nodes inside the larger component. As such, this is usually only computed for the nodes inside the larger component, since these are the ones that are generally closer to all the nodes, on average. In Equation 2.3 I denote Cl_i as the closeness of node i, N as the total number of nodes in the graph, and $d(u,i)$ as the length of the shortest path between the node s and i.

$$Cl_i = \frac{N-1}{\sum_{s \in N(G) \setminus i} d(s,i)} \tag{2.3}$$

Eigenvector centrality

The main idea behind *eigenvector centrality* (Bonacich, 1987) is that the power and status of a node are defined by the power and status of its neighbours. This measure takes into account both the number of neighbours of a node and how well connected those neighbours are. Equation 2.4 is the eigenvector centrality's formula. In this equation, ec_i is the eigenvector centrality of node i, x_{ij} is the entry on the i-th row and j-th column of the adjacency matrix X, and λ is the largest eigenvalue of X. d_i and d_j are the centralities of nodes i and j.

$$ec_i = d_i \frac{1}{\lambda} \sum_{j=1}^{n} x_{ij} d_j \tag{2.4}$$

Local clustering coefficient

Watts and Strogatz (1998) proposed a *local clustering coefficient* which is based on the property of transitivity found in social networks. This property says that the friends of one agent are also likely to be friends with each other.

In Equation 2.5 I denote the local clustering coefficient of node i as cc_i. e_{st} is the link that connects the nodes s and t, and $|e_{st}|$ is the proportion of links between the nodes within the neighbourhood of node i. k_i is the degree of node i. H_i is the neighbourhood of node i. $E(G)$ is the set of links in graph G.

$$cc_i = \frac{2|e_{st}|}{k_i(k_i-1)} \; : s,t \in H_i \, , \, e_{st} \in E(G) \tag{2.5}$$

2.1.3 Statistical measures for the network

Average degree

There are several statistical measures that attempt to describe and help us understand the structure of a network. Based on the degree of the nodes we can calculate the *average degree* of a network, which is simply the mean of the degrees of all the nodes in the network, and it can be used to measure the global connectivity of a graph.

Geodesic distances

The *geodesic distance* between two nodes is given by the minimum number of links needed to connect the two. If there isn't a path connecting them, then it is not possible to calculate this distance and it is conventionally defined as infinite. The *average geodesic distance* is the average of the geodesic distance for all combinations of pairs of nodes. If there are some pairs without a possible path between them, this measure cannot be computed and as such the *harmonic average geodesic distance* should be used since it turns the infinite distances into null distances. In Equation 2.6 hg^{-1} is the harmonic geodesic distance and $d_g(i,j)$ is the geodesic distance between nodes i and j. N is the number of nodes in the graph.

$$hg^{-1} = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i>=j} \frac{1}{d_g(i,j)} \tag{2.6}$$

Eccentricity, radius and diameter

Based on the geodesic distance one can also calculate the *eccentricity* of a node, which is given by the largest geodesic distance between the node and any other node in the network. This notion serves as the basis for two other network-level statistical measures:

radius and *diameter*. The radius of a network is defined by the minimum eccentricity of all the nodes in the network, while the diameter is defined by the maximum eccentricity found in the network. If a network is very sparse, its diameter will likely be high as there are less paths connecting distant nodes. This indicates how far two nodes in the network can be, in the worst case.

Density

The *density* of a network indicates the general level of connectedness in a network. A network can be described as sparse when it has a low density, and as dense when it has a high density. This metric is calculated as the proportion of links that exist in the network in comparison to the maximum possible number of links the network can have. In Equation 2.7, $\rho(G)$ is the density, E represents the number of edges in the network and $E_{max(G)}$ represents the maximum possible number of links in graph G. This maximum number is $\frac{n(n-1)}{2}$ for undirected networks and $n(n-1)$ for directed ones.

$$\rho(G) = \frac{E}{E_{\max(G)}} \qquad , 0 < \rho < 1 \tag{2.7}$$

Global clustering coefficient

The *global clustering coefficient* is another metric, and it can be calculated in several ways. Watts and Strogatz (1998) calculates the local clustering coefficient by computing it for each node and calculating their average.

2.1.4 Properties of real-world networks

According to Newman (2003a), most real-world networks have properties that cannot be found in random graphs. The ones I discuss here are the *small-world effect*, *transitivity*, *degree distributions*, *mixing patterns* and *community structure*, for they seem more relevant to the study developed.

Small-world effect

The *small-world effect* property was demonstrated by Travers and Milgram (1969). This property is found in social networks. In the experiment they carried out it was shown that distant individuals in a network are in fact connected by a small number of acquaintances. In mathematical terms this means that as the network size increases with a fixed mean degree, the average geodesic distance between pairs of nodes increases at a slower pace.

Transitivity

Newman (2003a) defines *transitivity* – sometimes called *clustering*, though it should not be confused with community detection – as a property related to sets of three nodes where at least one is connected to the other two. Oliveira and Gama (2012) add that when this happens there's an increased chance of the other two nodes being connected to each other, and so this property measures the density of triangles. In a social network this might mean that if a person has two friends they are likely to be friends as well. On a network where the nodes are connected by affinity, this might mean that if a person is similar to two others, then those two might also be similar to each other.

Degree distributions

As defined by Oliveira and Gama (2012), *degree distribution* is the probability distribution of the degrees of nodes on the network. If one considers $P(k)$ to be that distribution, then it represents the probability of a randomly chosen node having degree k. In random graphs, this distribution is rather homogeneous, with most nodes having similar degree. Barabasi and Albert (1999) found out that the same does not hold true for most real-world networks. Instead, what we find are highly skewed distributions where a majority of nodes has low degree and only a few nodes have high degree. This is defined as a *power-law* distribution, and networks that follow this distribution are called *scale-free networks*. If a network expands continuously and these new nodes have a preference for attaching to

well connected nodes, the network is likely to become a scale-free network.

Mixing patterns

Mixing patterns are related to how the nodes in the network are linked to each other. Nodes can have different characteristics or belong to different groups, and the way these characteristics affect the links depends on the network being observed. In the example by Newman (2003a), a network representing an ecosystem where species eat other species, we can have 3 groups of species: herbivores, carnivores and plants. Herbivores will be linked to plants and carnivores, but they'll rarely be connected to other herbivores. On the other hand, if we consider a social network representing friendships instead, we may find that people from the same age group are more likely to be friends with each other than with people from other age groups. Newman (2003b) calls this selective linking in social networks *assortative mixing*.

Community structure

According to Newman (2003a), the mixing patterns property suggests the existence of community structure in networks, which has always been assumed to exist. A community can be defined as a group of nodes that are highly connected between themselves but with much fewer connections to nodes outside of the group. This concept is similar to what we observe with mixing patterns, but the existence of assortative mixing does not guarantee the presence of a community structure. Nonetheless, it can be interesting to study the existence of these structures, and there are several algorithms that attempt to do this.

2.2 Community Detection in networks

2.2.1 Introduction

Real-world networks have underlying structures that govern the interactions between the nodes. These structures are present not only in social networks but also in many other fields, such as computer science, biology, among others. This organisation is usually of groups of nodes that have something in common. As defined by Newman (2003a), *communities* are groups of nodes with a high density of links within the group but a low density of links to other groups. These communities can tell us a lot about the underlying structures of a network, and as such should be given particular attention. Since most of the time these communities are not known in advance, one must use *community detection algorithms* to discover them.

2.2.2 Hierarchical clustering

Hierarchical clustering techniques are very popular because they do not require assumptions regarding number of clusters or size of clusters. The goal is to classify individuals into homogeneous groups according to their similarity or dissimilarity by building a hierarchical structure iteratively, revealing a multi-level structure where we can see the nodes by themselves as clusters all the way to a single cluster containing all the nodes (Oliveira and Gama, 2012). This property makes these methods very useful for situations where little is known about the structure of the communities, since one can see the different layers of the structure in a dendrogram and choose which one fits the reality of the network best. For example, in Figure 2.2 is a dendrogram where each letter represents a node. If we look at it from the bottom up, lines represent the sequential merging of nodes into a cluster. Alternatively, if seen from the top down it can be interpreted as the division of clusters into smaller ones until we have the individual nodes.

According to Kaufman and Rousseeuw (2005), hierarchical clustering techniques can

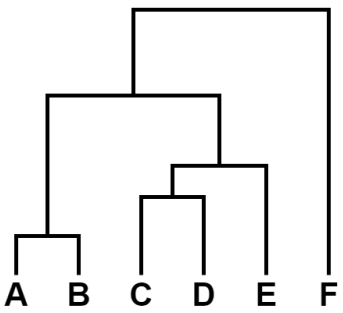

Figure 2.2: Example of a dendrogram.

be divided into two main groups, depending on how they build the structure of the hierarchy: *agglomerative* and *divisive*. In agglomerative algorithms, at each iteration the two most similar clusters are merged together. The *Walktrap* method belongs in this group. In divisive algorithms, clusters are split by removing the links connecting dissimilar nodes. These methods can then be seen as opposites, as the agglomerative algorithms join clusters while the divisive algorithms separate them. Agglomerative algorithms are bottom-up, starting from the nodes as clusters and ending with a single cluster. Divisive algorithms are top-down, starting from one cluster and ending with each node as an individual cluster.

The first step in a hierarchical clustering method is to define the dissimilarity measure that will be used as the distance between nodes. There are several measures to choose from, such as the Euclidean or Hamming distance. From this point forward both categories have different steps, which I go over next.

Divisive methods

Divisive methods have a criterion to remove the links that connect clusters. There are several divisive hierarchical clustering methods, with the most popular one being *Girvan-Newman's algorithm* (Girvan and Newman, 2002; Newman and Girvan, 2004). According to Girvan and Newman (2002), this method is also called *Edge Betweenness* because it targets the edges that are between communities, while agglomerative methods focus on

finding the most central links in communities. These authors define the *edge betweenness* of a link as the number of shortest paths between pairs of nodes that pass through it. Since communities do not have as many links between themselves as they have inside, then the shortest paths between nodes of different communities will have to go through the same few links, and as such these links will have a high edge betweenness. Their method targets these links and removes them in order to separate the communities. A dendrogram can be built from this process, allowing the user to view the order in which the clusters were divided. The user can then decide which partition of the hierarchy seems the most appropriate, or use a quality function to find the optimal partition.

Agglomerative methods

Agglomerative methods differ from divisive ones since they do not attempt to divide clusters, but instead merge them. In the case of agglomerative algorithms, after selecting the dissimilarity measure between the nodes, the second step is to select the dissimilarity measure between the clusters. Since clusters have many nodes, we can select any node from each of them and decide that the distance between them represents the distance between the clusters they belong to. There are several alternatives, such as calculating the middle point of all the nodes in the cluster (known as centroid) and compare that to the centroid of the other clusters (*centroid linkage*). We can also choose the pair of nodes (one from each cluster) that is the closest to each other and consider the distance between them to represent the distance between the clusters they belong to (*single linkage*). Other popular aggregation indices are *complete linkage*, *average linkage* and *Ward linkage* (Kaufman and Rousseeuw, 2005).

Having decided on these two parameters, the algorithms will calculate these distances. At each iteration, the closest clusters will be aggregated. The process will continue until there is only one cluster, composed of all of the nodes. A dendrogram can also be built in order to see how the clusters were merged and decide on the most adequate partition of the network.

2.2.3 Walktrap method

The Walktrap method (Latapy and Pons, 2006) is an agglomerative hierarchical clustering method. It uses the concept of *random walks* to calculate the distance between two nodes. This distance is used as the dissimilarity measure between nodes, and the dissimilarity measure between clusters is Ward linkage. This method is described further in Chapter 3.

2.2.4 Louvain's method

Louvain's method (Blondel et al., 2008) is an optimisation method that maximises *modularity* (Newman and Girvan, 2004). This is a measure of the quality of a partition of the network. This method has two phases that are repeated iteratively until a maximum value of *modularity* is achieved. This algorithm is explained in better detail in Chapter 3.

2.2.5 Infomap

Infomap (Rosvall and Bergstrom, 2007, 2008; Rosvall et al., 2009; Bohlin et al., 2014) takes on a different approach from the already mentioned algorithms in terms of concept, but in practise it shares some similarities. Infomap can be defined as a flow-based method, with heavy foundations on information theory, because it assumes that networks carry a flow. In practice, it uses random walks like the Walktrap algorithm and the two phase method used by Louvain's method. However, instead of maximising *modularity*, it minimises a *map equation*. This algorithm is further described in Chapter 3.

2.2.6 Evaluation measures

One focus of this study is to compare different community detection algorithms in regards to the effectiveness and balance of the communities generated. Measuring the effectiveness of the algorithms is not a trivial problem. There are many objective measures for the accuracy of a community detection algorithm, but those are mostly for the cases where

the real structure of the network is known. Steinhaeuser and Chawla (2010) present some of these metrics, namely *accuracy*, *Rand Index* (Rand, 1971) and *Adjusted Rand Index* (Hubert and Arabie, 1985). For the cases where the real structure of the network is unknown, the most popular measure is *modularity* (Newman and Girvan, 2004). *Modularity Density* is an alternative by Chen et al. (2013) that attempts to overcome some known flaws of *modularity*. These two measures are explained in more detail in Chapter 3.

For unsupervised tasks it is not uncommon to use the researchers' own judgement and feedback from experts on the matter to ascertain the quality of a partition. These measures are subjective and thus may be biased, but they can be useful since a mathematical equation does not take into account the real-world context of the network.

2.3 Information Retrieval

Information Retrieval (IR) is a very broad term, but according to Manning et al. (2008) it is defined as the finding of material that does not have a structured nature, which satisfies a need of information. It is concerned with the search, manipulation and representation of large collections of data, mostly text data (Büttcher et al., 2010). The most common usage is to find information from text documents that satisfy queries from the user, and these documents are usually stored on computers. IR is a very extensive field, so in this section I do a very brief overview of the concepts that are relevant to the study carried out.

The most popular IR services are web search engines like Google and other general search engines – for example, to find documents in your computer – but there are many other applications. According to Büttcher et al. (2010) it can be used in reverse, and with a given document find a set of keywords that can be useful to the user. This is used in news aggregation applications, which categorise new articles into already established tags, such as "business" and "sports". Another application are summarisation systems, which reduce the documents to a few keywords or sentences. Information extraction systems attempt to discover specific information in a document, such as locations and dates. There are many

other applications, which tend to cross several fields of study, mainly library, information science and computer science.

According to Büttcher et al. (2010), there is a basic architecture and structure present in most IR systems. The process starts from the user, who has a need for certain information. He creates a query with the terms he finds relevant to find that information. These terms can be words, numbers, dates, and so on. The query is then processed by the search engine, which will return a ranked list of results to the user. The search engine has the important task of maintaining and manipulating an inverted index for a document collection. This index is the central data structure in any IR system, providing a mapping between the terms in the document collection and where they appear. The search engine also gathers statistics that it uses to create the rankings according to relevance by calculating a score for each document. This task of attributing scores to documents depending on the user's query is the most fundamental problem in the field.

2.3.1 Terms and tokens

A *token* usually corresponds to a sequence of alphanumeric characters (A to Z and 0 to 9). These could also include structural information such as XML tags though. *Tokenisation* refers to the conversion of each document into a sequence of tokens. This is a critical step in the creation of the inverted index. Tokens are the link between the queries by the user and the documents. When the user makes a query, the tokenisation is applied to the query and those terms are matched against the index (Büttcher et al., 2010).

The documents in the document collection are pre-processed in order to create tokens. There are many processes that can be applied to the text, but I go over the most common and simple ones.

Punctuation It is usual to remove punctuation from the text as it is not important in many cases. However, sometimes this may not provide the best results when the punctuation makes a difference in the meaning of the sentence. For example, "I'll"

and "Ill" have very different meanings (Büttcher et al., 2010).

Capitalisation It is normal to normalise the case of the letters in the documents, usually turning them all to lower case. An example of why this is necessary is the capitalisation of the first letter in a sentence. For example, "Group" and "group" are essentially the same word and should be considered as such, and to do so we convert the upper case letter into lower case. This may not always be a good strategy since some words are only distinguished by the capitalisation. For example, "US" and "us" refer to different things (Büttcher et al., 2010).

Stemming This process is about reducing the terms to a *root form*. For example, "running" and "runner" can be reduced to their root form "run". The index must have a dictionary to perform this task (Büttcher et al., 2010).

Stopping *Stopwords* mainly consist of *function words*, which refer to terms that do not have a well-defined meaning by themselves. These include prepositions, articles, pronouns, articles and conjunctions. For example, "the" and "of" are considered *function words*. IR systems usually define a list of these stopwords, which are removed from the query. This should not have a negative impact on the effectiveness of the retrieval, though there are some exceptions (Büttcher et al., 2010).

Accents and diacritics English does not have many of these, but other languages do. However, regardless of the language most users generally do not type these diacritics, so removing them is usually the best approach (Manning et al., 2008).

Numbers In some cases, numbers may also be removed if they are not relevant to the IR system.

2.3.2 Term weighting

Another function of the search engine is to create a ranking of the documents that match the query of the user. Assuming a *bag-of-words* model – in which the order of the words

does not matter – we can attribute weights to the terms in the documents based on the statistics of the occurrence of those terms.

Tf-idf weighting

One of the most common weighting schemes is the *tf-idf*. The *tf-idf* of a term t in document d is given by Equation 2.8. $tf_{t,d}$ is the term frequency of term t in document d, which is simply the number of occurrences of the term in the document. The idf_t is the inverse document frequency of the term in the document collection (Büttcher et al., 2010).

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t \tag{2.8}$$

The document frequency is the number of documents that contain the term t. However, common terms would be favoured by this measure, so the inverse document frequency is used to give rare terms a higher score. The idf_t is given by Equation 2.9, where N is the number of documents in the collection.

$$idf_t = \log\left(\frac{N}{df_t}\right) \tag{2.9}$$

Given this, the $tf\text{-}idf_{t,d}$ will be:

- Higher when the term appears many times but in a small number of documents. This will give a higher discriminating power to these documents.
- Lower when the term doesn't appear many times in a document or appears in many documents.
- Lowest when the term appears in almost all of the documents in the collection.

With this we can create a vector for each document that has the *tf-idf* of each of the terms in the document collection. Based on this vectors we can then calculate the similarity between two documents, or between a document and the user query. In Section

3.5 I present the *cosine similarity*, which is used to calculate the similarity between two documents.

2.4 Affinity Miner

Affinity Miner (Brazdil et al., 2015; Trigo et al., 2015) is an information retrieval tool that attempts to facilitate the discovery of researchers by other researchers and investigation centres.

Affinity Miner uses data from articles written by the researchers. The terms in the titles, keywords and abstracts of those publications are processed using bag-of-words and vector representation, and the usual preprocessing is applied by removing numbers, punctuation and other elements of the sort (Feldman and Sanger, 2007). The list of documents is then transformed into a *document-term matrix* where the columns contain all the terms in the set of articles and each row refers to a document. Each value of the matrix indicates the frequency with which the term on the column appears in the document on the row, using *tf-idf* weighting. The process does not end here, and is followed by the calculation of the *cosine similarity* for each pair of researchers. This is then used to create a network where each node is a researcher and the links represent similarity. From here, the community detection algorithm is applied to detect *affinity groups*. If the data has a variable with the group each researcher belongs to, Affinity Miner also generates these networks and communities for each group considered individually. This entire process is programmed mostly in `R` and its packages.

2.4.1 TextRank

Affinity Miner also generates relevant keywords that describe each author and affinity group. The user is able to search for researchers of interest, and the system returns the researchers that match those terms based on the keywords it generated. Another functionality is the ability to click on a researcher's node in order to see the keywords that

describe him and those that describe the affinity group he belongs to. These keywords are generated using the *TextRank* method (Mihalcea and Tarau, 2004).

TextRank is a graph-based ranking model for text processing. It uses "votes" as its basic idea. When one node in the graph connects to another it is casting a vote for that other node. The importance of the node casting the vote is also taken into account, as it will determine how important the vote is. The importance of a node is given by the votes cast for it and the importance of the nodes making those votes.

In the graph, the nodes and links are determined by the application. The nodes are words or other text entities, while the links are the relations between those nodes. The application of graph-based ranking algorithms to natural language texts follows the same steps regardless of the characteristics of the elements in the graph. The first step is to identify the text units that best define the task at hand and add them to the graph as nodes. The second step is to identify the relations between these text units, in order to create the links between the nodes in the graph. These links can be unweighted or weighted, and directed or undirected. When the graph is complete, the ranking algorithm iterates until convergence is achieved. Finally, nodes are sorted based on their final scores.

Keyword extraction

TextRank can be used for the task of *keyword extraction*. This is the automatic identification of a set of terms that best describe a document. These keywords have many applications, such as the classification of text or as a summary of a document.

This algorithm is fully unsupervised. It starts by tokenising the text and annotating it with speech tags, which will allow for the application of syntactic filters later on. At first it only considers single words as candidates to add to the graph, in order to prevent it from growing excessively. Then the units that pass the syntactic filter are added to the graph. Links are added between units that co-occur within a window of N words, which ranges from 2 to 10 words.

When this graph is constructed, it will be unweighted and undirected. The initial

score associated with each node is set to 1, and then the graph-based ranking algorithm previously described is run on the graph until it converges.

After each node has a final score, they are sorted in reverse order of their score. The top T nodes in this ranking are used in the post-processing phase. T is a value that is usually set between 5 to 20, but Mihalcea and Tarau (2004) considered it as one third of the number of nodes in the graph. In the post-processing phase, the keywords from the top T nodes are marked in the text. If these keywords appear adjacent to each other, they are collapsed into one multi-word keyword.

Chapter 3

Comparing Some Community Detection Methods

3.1 Motivation

Graphs representing networks are frequently used to represent entities and how they interact with one another. Biology, computer science, marketing, among other areas, take advantage of this representation to enhance our understanding of how real-world networks are structured and organised. In biology, for example, we can represent an ecosystem as a network and visualise the food chain present in it. Marketing can use social networks to illustrate the impact and diffusion of their campaigns throughout their customers and potential customers. Companies can use networks to visualise how they are organised and how information flows through the company.

Real-world networks have underlying structures governing the interactions between the agents. If we wish to better understand the network of interest, discovering these structures becomes a necessary task. One way to approach this is by looking for some observable effects of these structures: the formation of groups of agents that have a higher degree of interaction among themselves than with the rest of the network. In the context of social networks, we may regard these as groups of friends or communities of people.

Identifying these groups can be done with community detection algorithms. There are many algorithms developed for community detection, which focus on different aspects of what constitutes a "group". Some methods will perform better than others in different contexts. It becomes important to analyse and experiment with some of these methods in order to discover which ones are capable of providing the most adequate results for the given situation.

The objectives of this chapter are to find an adequate representation of the network for further analysis in Chapter 4 and identify the community detection algorithm which achieves better performances.

3.2 Organisation of this chapter

In Section 3.3 I briefly describe the tools used in this study, and in Section 3.4 I quickly summarise the data used. Following this, in Sections 3.5 and 3.6 I describe how the basic network for this project was created from the data. In Section 3.7 I discuss one method of pruning this network, why this is necessary and its implications. In Section 3.8 I describe the 3 community detection algorithms I used in this study, namely *Walktrap*, *Louvain's method* and *Infomap*. In the section after I discuss ways to evaluate the community structure identified by the previous algorithms, including both objective and subjective measures.

In Section 3.10 I look for a way to select the best values for the pruning method. I start by using objective measures, and then implement a new measure in order to obtain better results. After these values are found, in the subsequent section I compare the performance of the three community detection methods for the best values of the pruning method already identified.

3.3 Implementation details

The tools used in this study were `R` (R Core Team, 2017) and the package `igraph` (Csardi and Nepusz, 2006). Affinity Miner (Trigo et al., 2015) was used for the text processing tasks and network visualisation. I also modified its source code in order to do the experiments in this study.

3.4 Data used in this study

The data for this project was provided by the *Authenticus* team (Authenticus, 2017) and spans the period from January 1972 to December 2016. It contains information about the publications from researchers affiliated to FEP. The data used in this chapter has 66 researchers with 1149 publications. A more detailed description of the data can be found in Section 4.3.

3.5 Similarity between two documents

Using the data previously mentioned, I created a document for each author containing the titles of his publications. The titles are processed using bag-of-words and vector representation, and the usual text pre-processing is applied by removing numbers, punctuation, stopwords and lowering the case of the terms (Feldman and Sanger, 2007). The list of documents is then transformed into a *document-term matrix* where the columns contain all the terms with at least 3 characters in the set of documents and each row refers to a document/researcher. Each value of the matrix indicates the frequency with which the term on the column appears in the document of the researcher on the row, using *tf-idf* weighting (Manning et al., 2008).

The similarity between two documents is calculated using the *cosine similarity*. Tan et al. (2006) present this as a commonly used way to measure the similarity between two

vectors that represent terms used in documents. In this study it is used to calculate the similarity between the vectors of terms belonging to different researchers. Given two vectors X and Y, it can be calculated as expressed in Equation 3.1.

$$cos(X,Y) = \frac{\sum_{k=1}^{n} X_k * Y_k}{\sqrt{\sum_{k=1}^{n} X_k^2} * \sqrt{\sum_{k=1}^{n} Y_k^2}} \tag{3.1}$$

Since the document-term matrix uses *tf-idf* weighting to generate the vectors, the cosine similarity between two vectors will vary between 0 and 1 (Tan et al., 2006). A value of 1 indicates the two vectors are equal, while a value of 0 indicates they do not share any terms. I obtained an adjacency matrix with the researchers as rows and columns, where the element in row i and column j is the cosine similarity between researchers i and j.

3.6 Representing similarity matrix as a graph

The cosine similarity matrix obtained previously was converted to a network with 66 nodes, each representing a researcher, and 1595 links between them, each corresponding to the cosine similarity between the pair of nodes it connects.

3.7 Pruning links in a graph

Pruning links refers to the removal of less important links from a network to simplify it. This can be used when the number of links is unusually high. When using the cosine similarity measure, it is likely that it will assume values above 0 quite often. If two vectors have a term in common, this similarity would be above 0. This would then lead to a high number of links between nodes, which in turn could undermine the task of community detection: if all nodes were connected to each other, then the algorithms could have trouble distinguishing areas of high density of links and low density, which could

lead to the generation of few but very large communities that fail to represent the structure of the network. On top of this, it would be difficult to visually explore the network, since there would be an immense amount of links. The network on the left side of Figure 3.1 illustrates this.

In order to deal with this issue, one method is to apply a threshold, which I will refer to as the *cosine similarity threshold*. This acts as the minimum value for the cosine similarity, under which the links are disregarded. For a threshold of 0.05, any link with a cosine similarity below 0.05 will be given a weight of 0, effectively removing them from the network. The value of this threshold should be experimented with, since the number of links can greatly vary with it. Figure 3.1 shows the difference between not having a threshold and having a threshold of 0.05, for the same data.

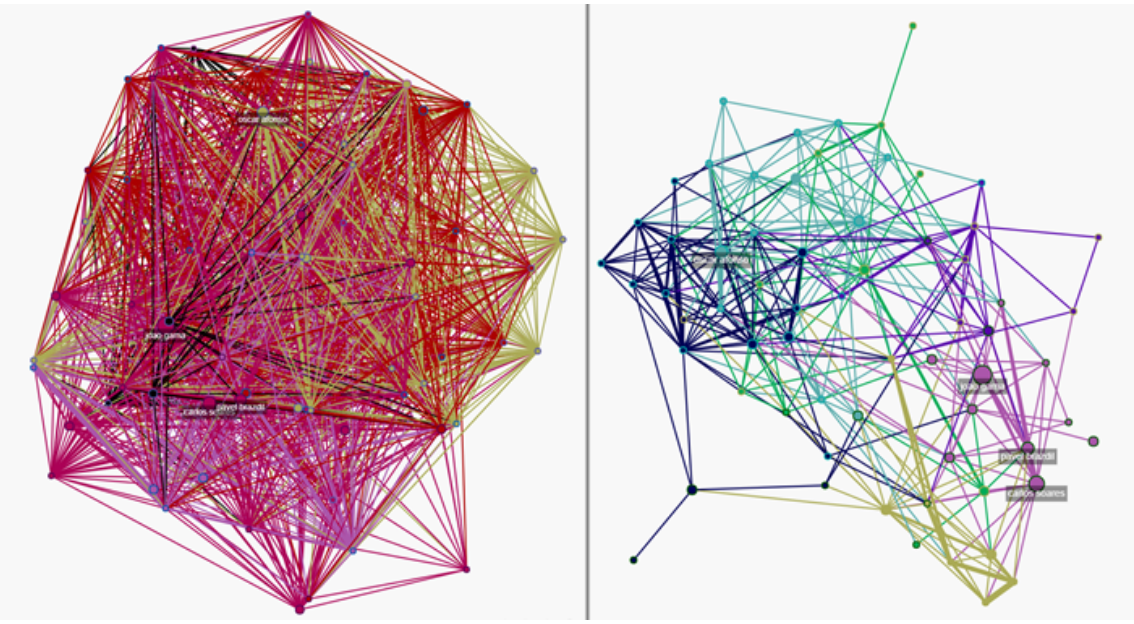

Figure 3.1: Threshold comparison: application of one community detection method on the same data but with different thresholds. Left side: no threshold. Right side: 0.05.

One downside of this method of pruning is that some nodes may be completely cut from the rest of the network. In Section 2.1.1 I have defined a *component*, which is a subgraph that is not connected to the rest of the network (Easley and Kleinberg, 2010). I also defined a bridge as a link that connects two subgraphs that would otherwise be

considered separate components. By applying this threshold, we are deleting links simply based on their weight. This can lead to the deletion of bridges, separating a network into two or more components.

3.8 Community detection algorithms

3.8.1 Walktrap method

The *Walktrap* method is an agglomerative hierarchical clustering method that also uses *random walks*. A *walk* is a sequence of nodes and links, which starts and ends with a node (Iacobucci, 1994). A *random walk* is a process according to which a random walker moves away from its starting point. For example, in the context of a network a random walker can be placed on a node and randomly walk between connected nodes. Since there are many links connecting the nodes, there are several paths the random walker can take to reach a certain node, and the probability of the random walker being at a certain node in a certain amount of steps differs depending on both the starting node and the final node considered. Latapy and Pons (2006) use these probabilities to calculate the distance between nodes, based on the definition of community: if a community is a group of nodes densely connected to each other and sparsely connected to nodes outside of the community, then the random walker will spend more time inside the community of the starting node. This means the probability of reaching a node in the same community as its starting node is higher than that of reaching a node outside. This distance is used as the dissimilarity measure between nodes in an agglomerative hierarchical clustering, and then Ward linkage is used as the dissimilarity measure between clusters. The rest of the process works like a usual agglomerative hierarchical clustering.

Ward linkage

Ward linkage (Ward, 1963) is used in clustering algorithms to measure the dissimilarity between two clusters. This method is based on the Euclidean distance between the clusters' centroids, multiplied by a factor. This is presented in Equations 3.2 and 3.3, where $|R|$ and $|Q|$ represent the number of nodes in clusters R and Q, respectively. $\bar{x}(R)$ and $\bar{x}(Q)$ represent the centroids of the clusters. $||\bar{x}(R) - \bar{x}(Q)||$ is the Euclidean distance between the two clusters. The actual value of this dissimilarity measure is $d(R,Q)$, the square root of Equation 3.2 (Kaufman and Rousseeuw, 2005).

$$d^2(R,Q) = \frac{2|R||Q|}{|R|+|Q|} ||\bar{x}(R) - \bar{x}(Q)||^2 \tag{3.2}$$

$$d(R,Q) = \sqrt{d^2(R,Q)} \tag{3.3}$$

3.8.2 Louvain's method

One can look at the community detection problem as the task of finding the optimal partition of the network into communities, or modules. Though it is currently computationally impossible to get this answer in a relatively fast amount of time, an optimised solution can be found via heuristics. Optimisation methods, such as *Louvain's method* (Blondel et al., 2008), focus on maximising an objective function. In this method, the objective function used is *modularity* (Newman and Girvan, 2004), which is a measure of the quality of a partition of the network. I will go over *modularity* in more detail in section 3.9.

As described by Blondel et al. (2008), this community detection algorithm iterates over two distinct phases. Assuming a weighted network, which is the case of the network I work with in this study, the algorithm starts with each node being its own community. For each node i it evaluates the gain in *modularity* that would occur if i were removed from its community and moved to the community of one of its neighbours. This node goes to the community that leads to the maximum gain in *modularity*, as long as this value is positive.

Otherwise, i remains in its original community. This process is repeated sequentially for all nodes, with the possibility of a node being considered several times. When no further improvement can be reached, which means a local maxima of the modularity was achieved and no individual move can improve this value, this phase ends. The results of this stage may be altered depending on the order considered for the nodes, and while preliminary results show this does not influence the *modularity* attained by much, they also indicate it influences the computation time.

When the first phase ends, the second phase starts. In this phase, the algorithm builds a new network using the previously discovered communities as the nodes. The links connecting the same two communities are merged together, and so the weight of the new links is simply the sum of the weights of the nodes that previously connected those two groups. Links inside a community now become self-loops. This ends the second phase. At this point, the first phase can be applied again. Blondel et al. (2008) call the two phases a *pass*, and this *pass* can be done over and over again, further reducing the number of communities until there are no more alterations and a maximum value of *modularity* is achieved. Since the algorithm joins communities at each *pass*, it has a hierarchical structure akin to what can be found in hierarchical clustering methods.

Blondel et al. (2008) concluded that their method performs very well against other community detection methods, while being significantly faster. However, they only tested the accuracy of the final communities returned by the algorithm, and as such the performance of the intermediary partitions of the networks still has to be studied.

3.8.3 Infomap

Infomap (Rosvall and Bergstrom, 2007, 2008; Rosvall et al., 2009; Bohlin et al., 2014) takes on a different approach from the already mentioned algorithms in terms of concept, but in practise it shares similarities with the previously described methods. While the previous algorithms focused on finding a structure in the network, Infomap assumes that networks carry a flow. Even if we have a static network, in reality it was generated dy-

namically. For example, if we look at a community of friends, there's information being exchanged between people. There is a flow in the way this information is spread along the network, and as such it is necessary to understand the structure of the network in regards to that flow, otherwise it is not possible to fully understand how the communities behave at the macro-level. Taking all this into account, Infomap can be defined as a flow-based method, with heavy foundations on information theory.

The basic idea behind Infomap is to be able to describe the network as efficiently as possible. To understand this better, Rosvall and Bergstrom (2007) provide an example. Let us assume a network X, a signaller who knows the full network X, and a signal receiver who does not. The goal of the signaller is to send the information about the network to the receiver as concisely as possible. The signaller must encode that information as a simple description, Y. This description is then sent to the receiver, who must now decode the message. However, since the information about the network was simplified, the receiver must make guesses, Z, about what the actual structure of the original network is. Better information means the receiver is able to reconstruct the network better, and thus has to make less guesses about the structure since there are less unknown elements. Obviously, the ways to summarise the information about the network are endless. The one chosen by Rosvall and Bergstrom (2007) based itself on information theory, according to which the best description Y of the network X is the one that maximises the mutual information between the two.

The Infomap algorithm is based on the *map equation* (Rosvall et al., 2009). This uses an idea similar to that of the Walktrap method, since it also takes advantage of the likelihood of a random walker staying trapped in a community during his random walk. Here, the random walker represents the flow of the network, and the aim is to describe the path taken by the random walker as concisely as possible. This description will be dependent on how well the network is partitioned, since there can be partitions that result in shorter description lengths and others that result in longer lengths. According to Bohlin et al. (2014), "the partition with the shortest description length is the one that best captures

the community structure of the network with respect to the dynamics on the network". By minimising the length of the description, this map equation is able to reveal important characteristics of the network structure.

The rest of the Infomap algorithm works similarly to Louvain's method. As described by Bohlin et al. (2014), in the first step each node constitutes its own cluster. In the first phase each node is, in a random sequential order, moved to the neighbouring cluster that leads to the largest decrease of the map equation. If the map equation can't be reduced for that node, it stays in its own cluster. This is repeated with new random sequential orders until no more decreases of the map equation can be found. After this comes the second phase, where the previously created clusters now constitute the nodes of the network, and the first phase happens again. These two phases are repeated one after the other successively until the map equation can't be decreased further. Up to here, the main difference between Louvain's method and Infomap is the objective function, since the first attempts to maximise *modularity* and the second attempts to minimise the map equation.

Bohlin et al. (2014) add that this algorithm can be improved further. In the phases explained before, nodes never leave a community to join another after they have been merged. If one breaks this rule and allows nodes or even entire parts of the clusters to move freely, the accuracy can be improved.

3.9 Evaluating communities

3.9.1 Modularity

The concept of *modularity* was introduced by Newman and Girvan (2004) and it measures the fraction of links in the network that connect nodes from the same community minus the fraction of links going out of that community (Equation 3.4, Chen et al. (2013)).

$$Q = \sum_{c_i \in C} \left[\frac{|E_{c_i}^{in}|}{|E|} - \left(\frac{2|E_{c_i}^{in}| + |E_{c_i}^{out}|}{2|E|} \right)^2 \right] \tag{3.4}$$

The notation is as follows:

– C: set of all communities;

– c_i: specific community in C;

– $|E_{c_i}^{in}|$: number/sum of weight of edges between nodes in community c_i;

– $|E_{c_i}^{out}|$: number/sum of weight of edges from community c_i to nodes outside of it;

– $|E|$: number/sum of weight of all edges in the network;

Modularity can have both positive and negative values, though it is always smaller than 1. If the *modularity* is negative or zero, it suggests the graph has no community structure. The larger the value of *modularity* the better, as it means the number of links inside a community is bigger than what would be expected if the same community had random links. *Modularity* is often used as a quality function by community detection algorithms, so it should be taken into account that evaluating network partitions with this measure may benefit the algorithms that optimise it.

Modularity is meant to compare the partitions of the same network obtained with different community detection algorithms. It is not supposed to compare results obtained with different networks, though it can be used when comparing networks of similar sizes. By deleting links from a network, we are effectively working with different networks, so *modularity* may not be an adequate indicator in such a situation.

Fortunato and Barthélemy (2007) show that *modularity* has a resolution limit, an intrinsic scale which depends on the degree of interconnectedness between pairs of communities and the total number of links in the network. If a community has a size smaller than this limit, the community detection algorithm may not find it. It is impossible to determine whether a community is a single module or a cluster of smaller modules of a size below this resolution limit. Though Fortunato and Barthélemy (2007) only proved this for unweighted and undirected networks, Berry et al. (2011) derived the extension of this argument to weighted networks.

3.9.2 Modularity density

Chen et al. (2013) point out the two problems of *modularity*: in some cases it favours small communities, while in other cases it favours large ones. To address these issues they propose a new metric called *modularity density* (Q_{ds}). They introduce the *modularity with split penalty*, which attempts to reduce the favouring of small communities by considering the links connecting different communities. The *split penalty* is defined as the fraction of links that connect nodes of different communities, and can be found in Equation 3.5. This value is then subtracted from the *modularity*. The idea behind this is that *modularity* is concerned with the positive effect of grouping nodes together by considering the links between them, while the *split penalty* focuses on the negative effect of ignoring links between nodes of different communities. However, this only worsens the resolution limit problem. To solve this, Chen et al. (2013) created the *modularity density*, as presented in Equation 3.6, which incorporates the number of edges and number of nodes in the communities into the *modularity* function and into the *split penalty*. Equations 3.4, 3.5 and 3.6 are for undirected networks, but Chen et al. (2013) also present the formulas for directed networks. These equations can be used for undirected networks, both unweighted and weighted. The difference is that unweighted networks use the number of edges, while weighted networks use the sum of the weights of the edges.

$$SP = \sum_{c_i \in C} \left[\sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{|E_{c_i,c_j}|}{|E|} \right] \tag{3.5}$$

$$Q_{ds} = \sum_{c_i \in C} \left[\frac{|E_{c_i}^{in}|}{|E|} d_{c_i} - \left(\frac{2|E_{c_i}^{in}| + |E_{c_i}^{out}|}{2|E|} d_{c_i} \right)^2 - \sum_{\substack{c_j \in C \\ c_j \neq c_i}} \frac{|E_{c_i,c_j}|}{2|E|} d_{c_i,c_j} \right] \tag{3.6}$$

The new notation is as follows:

- $|E_{c_i,c_j}|$: number/sum of weight of edges connecting community c_i to community c_j;
- $|c_i|$: number of nodes in community c_i;

– d_{c_i}: internal density of community c_i (Equation 3.7);

– d_{c_i,c_j}: pair-wise density between communities c_i and c_j (Equation 3.8).

Note that when calculating both d_{c_i} and d_{c_i,c_j} (equations 3.7 and 3.8, respectively), $|E^{in}_{c_i}|$ and $|E_{c_i,c_j}|$ are the number of edges for both unweighted and weighted networks and not the weights.

$$d_{c_i} = \frac{2|E^{in}_{c_i}|}{|c_i|(|c_i|-1)} \tag{3.7}$$

$$d_{c_i,c_j} = \frac{|E_{c_i,c_j}|}{|c_i||c_j|} \tag{3.8}$$

3.9.3 Subjective quality

Objective indicators may not always adapt to every context, specially when they are applied to a scenario which they were not designed for. A subjective attribute may be used to determine whether or not they are performing adequately.

I created a *subjective quality* attribute, Q_s, to verify this. Given the context of the network, which is of researchers from the School of Economics, I would expect the network to be composed of balanced groups. I would not expect to find many groups of only 1 or 2 researchers, nor would I expect to find very large groups encapsulating most of the researchers. Having only one or two communities with very few researchers is not necessarily negative. Sometimes researchers do not share any similarities. However, having too many communities of this size is not likely to be an accurate representation of the organisation of the Faculty. I would expect to see communities of balanced sizes, containing 4 to 20 researchers. Above this value the algorithm is likely placing most authors in one big community, and this does not tell us much about the structure of the Faculty. The values presented served as loose guidelines for me to assign a value of Q_s to each network.

Q_s can assume 5 values: 0.2, 0.4, 0.6, 0.8 and 1. A value of 1 is attributed to the seemingly best results, while a value of 0.2 is attributed to the worst results. The rest of

the values fall between that range. Table 3.1 has some examples of attributed Q_s values. The two first examples have the highest Q_s. The first contains no communities below a size of 4, and none above 20. The second contains two communities of only 1 researcher, but as mentioned before this is not necessarily negative. The rest of the communities in this example are well balanced. The last two examples have the lowest Q_s. One of them only has 1 community of 66 researchers, which does not tell us anything about the structure of the Faculty. The last one has too many small communities, with 10 of them being below a size of 4.

Community Sizes	Q_s
4, 7, 7, 7, 11, 12, 18	1
1, 1, 7, 8, 9, 10, 10, 10, 10	1
1, 1, 6, 6, 11, 15, 26	0.8
1, 1, 2, 3, 4, 6, 6, 7, 8, 8, 20	0.6
3, 6, 7, 11, 39	0.4
66	0.2
1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 4, 5, 6, 6, 7, 7, 8, 8	0.2

Table 3.1: Examples of sizes of communities and their attributed *subjective quality* value.

A downside of this indicator is that it is heavily influenced by my own bias. To account for this, a possible solution would be to request researchers and other people with a good knowledge of the Faculty to attribute scores to each example, and derive a single rating from those scores. This rating would be the new Q_s.

3.10 Selecting the cosine similarity threshold

To determine the best values of this threshold, I started by calculating the *modularity* and *modularity density* of the communities obtained with different thresholds and three different algorithms – Walktrap, Louvain and Infomap. The interval of the threshold begins at 0.01, which is equivalent to not having a threshold since there are no links with a similarity value below 0.01. At the other end I have used 0.6. Figure 3.2 shows that both *modularity* and *modularity density* tend to increase with the cosine similarity threshold.

This suggests that higher thresholds lead to better partitions of the network. However, this is not the case. The *modularity* reaches its peak for the threshold of 0.36, while the maximum of the *modularity density* is reached for the threshold of 0.53, at which point both indicators become nearly identical. For a threshold of 0.36 the network has only 19 links between its 66 nodes, leading to 49 communities of only 1 researcher. Each of these 49 researchers is completely disconnected from the others, that is, each of them is a separate component. This does not seem right. For the Louvain and Infomap methods the plots showed an identical trend, with the maximum values being exactly the same.

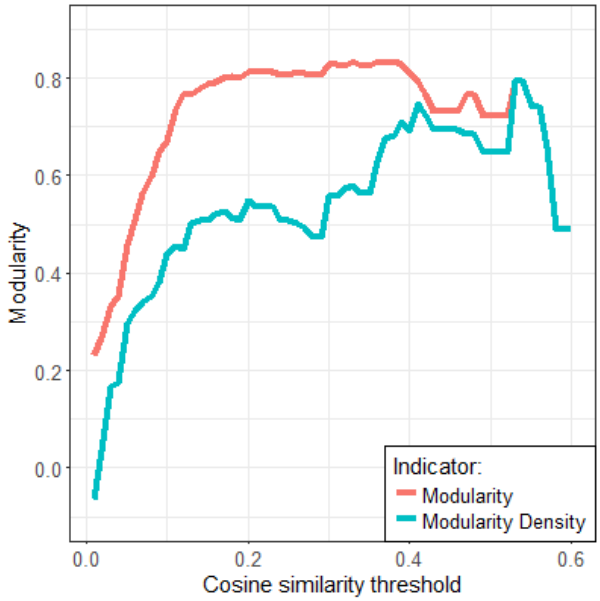

Figure 3.2: *Modularity* and *modularity density* versus the cosine similarity threshold, using Walktrap's method for community detection.

This result is, in a way, expected since *modularity* is not meant to compare partitions of the data obtained from different networks. By deleting links, we are effectively changing the size of the network. Though *modularity* can be used to compare networks of similar sizes, I considered here very different sizes. For example, a network with a threshold of 0.01 contains 1595 links, while if the threshold is increased to 0.1 the number of links goes down to 99. To my best of knowledge, there is no pointer in the literature that we could use for a case like this. As such, I decided to modify the *modularity* by introducing

a penalty for the number of components.

3.10.1 Modularity with component penalty

Previous results suggest that *modularity* is not the best measure to assess the results of community detection algorithms when networks of different sizes are involved. As such, I decided to use a penalty that is derived from the number of components in the network and subtract it from the value of *modularity*. This is because by increasing the cosine similarity threshold we often cut bridges that lead to a higher number of separate components, which is an undesirable effect.

I have decided to use the function $log_2(n)$ for this penalty, where n represents the number of components. Using a log_2 means that networks with only 1 connected component – no nodes disconnected from the rest of the network – have no penalty. The distribution of this value by cosine similarity threshold follows a shape somewhat similar to that of the *modularity*, as can be seen in Figure 3.3.

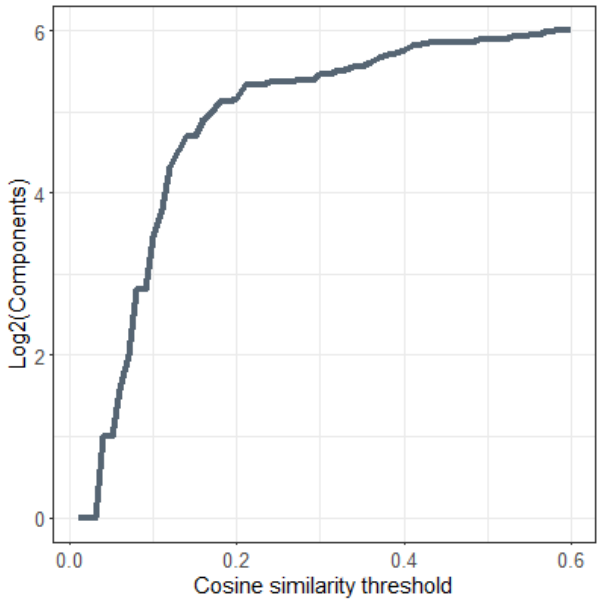

Figure 3.3: log_2 of the number of components versus the cosine similarity threshold, using Walktrap's method for community detection.

My modified measure is referred to as Q_p, *modularity with component penalty*. It is defined in Equation 3.9, where w_p denotes the weight of the penalty term and n denotes the number of components in the network. Q_m denotes *modularity*, which can be the original *modularity* or *modularity density*. I have used both of these variants in subsequent tests.

$$Q_p = Q_m - w_p * log_2(n) \tag{3.9}$$

My aim is to find an interval of values for the cosine similarity threshold which return the best results. First I had to determine what constitutes "good" results, and for that I resorted to the *subjective quality*, Q_s. Then I did a grid search to select which *modularity* to use and the value of the weight of the penalty, w_p. I concluded that the original *modularity* performed better than the *modularity density*, since the latter was not able to separate the "good" results from the "bad" ones, regardless of the value of w_p. For this weight, the value of 0.3 appears to provide the desired outcome. Though not perfect, this value allow us to find an interval of thresholds where the "good" results are concentrated. On top of that, it seems to create an easy to interpret border: when Q_p is above 0 the results are often good, but when it is below 0 the results are usually bad.

3.10.2 Selecting the best threshold interval

Using the *modularity with component penalty*, I was able to have a better overview of the most adequate thresholds for further analysis. Table 3.2 shows some of the results obtained with this formula. The *subjective quality* seems to generally move in the same direction as the *modularity with component penalty*, suggesting this is a satisfactory indicator. Figure 3.4 shows that the results labelled with a higher *subjective quality* attribute are all in the interval of *modularity with component penalty* above 0 and a cosine similarity threshold equal to or below 0.06. Based on this, I can determine that thresholds between 0.01 and 0.06 seem to generally provide the best partitions of the network, with a few exceptions depending on the algorithm used.

These exceptions come from the *Infomap* algorithm and the thresholds of 0.01 and 0.02, which obtain a *modularity with component penalty* score of 0. This happens because they only detect 1 community, so their *modularity* is 0 and the logarithm of base 2 of the number of components is 0 as well.

It is not a good idea to select the best threshold based on this indicator however, as it is not very rigorous and may be influenced by my own bias, and as such I considered the interval between 0.01 and 0.06 (inclusive) as adequate values for the cosine similarity threshold, instead of choosing the best value.

Rank	**Algorithm**	**Thresh.**	**Mod.**	n	**Community Sizes**	w_p	Q_p	Q_s
1	Louvain	0.03	0.352	1	4, 7, 7, 7, 11, 12, 18	0.3	0.352	1
2	Walktrap	0.03	0.332	1	1, 1, 6, 6, 11, 15, 26	0.3	0.332	0.8
3	Louvain	0.02	0.305	1	4, 7, 7, 7, 10, 13, 18	0.3	0.305	1
14	Louvain	0.06	0.523	3	1, 1, 7, 8, 9, 10, 10, 10, 10	0.3	0.048	1
17	Infomap	0.02	0.000	1	66	0.3	0.000	0.2
19	Louvain	0.07	0.579	4	1, 1, 1, 4, 7, 7, 10, 10, 11, 14	0.3	-0.021	0.6
20	Walktrap	0.07	0.564	4	1, 1, 1, 1, 2, 2, 3, 5, 7, 8, 8, 9, 18	0.3	-0.036	0.4
21	Infomap	0.07	0.563	4	1, 1, 1, 2, 2, 2, 4, 4, 5, 6, 6, 7, 7, 18	0.3	-0.037	0.2

Table 3.2: Indicators analysed for some combinations of algorithm and threshold, ordered by their Q_p.

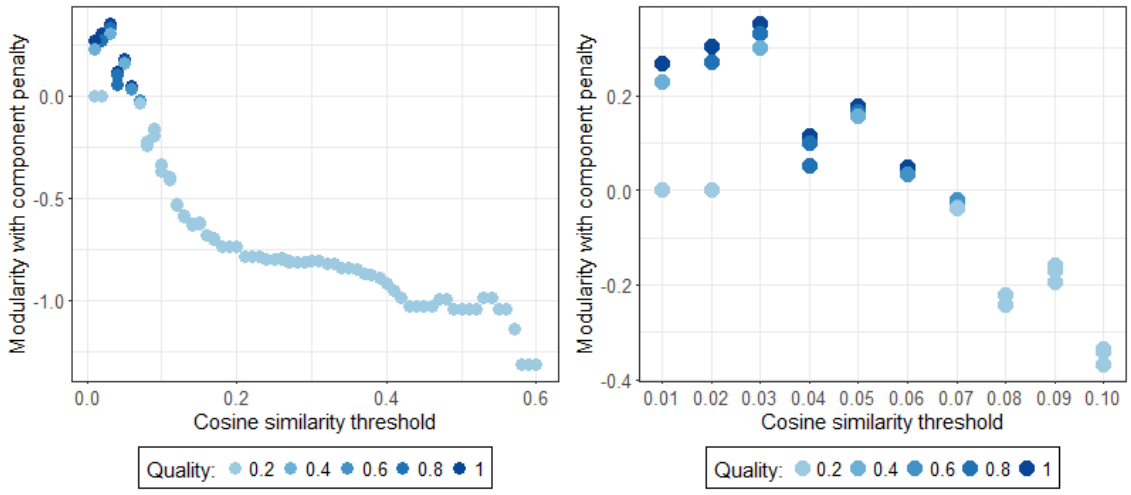

Figure 3.4: *Modularity with component penalty* (Q_p) versus threshold, coloured by the *subjective quality* attribute (Q_s). The plot on the left includes all values of the threshold, while the one on the right focuses on the thresholds between 0.01 and 0.1.

3.11 Comparing the three community detection methods

Using modularity and modularity density

Having selected the best threshold interval, I looked at the community detection algorithms to discover which one seemed to perform better in this interval. As before, I cannot compare all the combinations of threshold and algorithm with each other. In this case however I can compare the three algorithms for each threshold value individually, because by using the same threshold I am using the same network. As such, *modularity* and *modularity density* are the most adequate indicators in literature to evaluate the performance of the algorithms. However, these favour different algorithms: *modularity* favours Louvain's method, while *modularity density* favours Infomap. This can be seen in Figure 3.5.

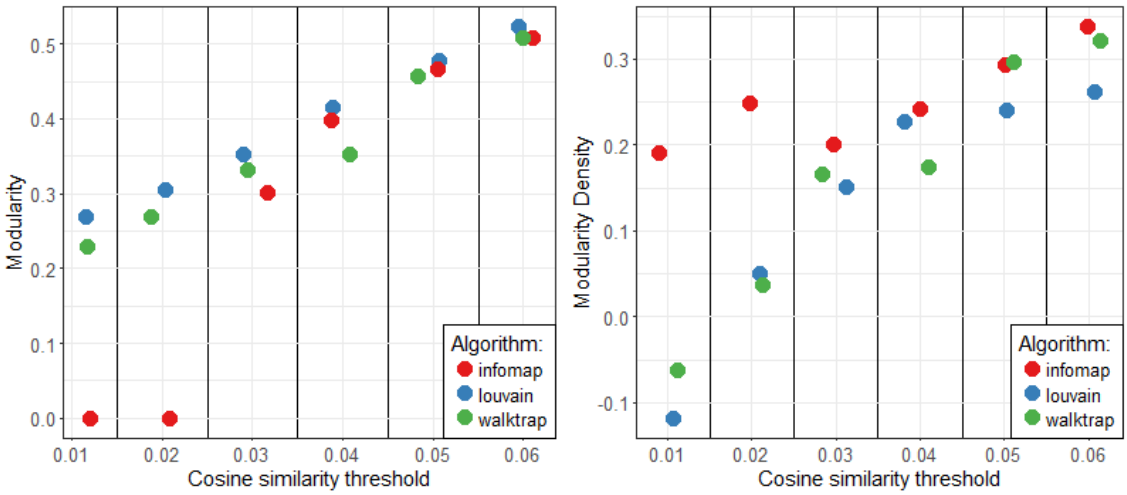

Figure 3.5: *Modularity* (left) and *modularity density* (right) versus threshold, coloured by the algorithm used. Some jitter was added to the points to allow for the visualisation of overlapping points.

I then used the previous *subjective quality* attribute to figure out if any of these indicators was in line with my subjective evaluation by plotting the same points in Figure 3.5 by *subjective quality* instead of *algorithm*, obtaining Figure 3.6. Here we can see that *modularity* has higher values when the *subjective quality* is higher, but for the *modularity density* the opposite happens. Table 3.3 contains this information as well. This suggests

that *modularity* is a more appropriate indicator in this context. It seems capable of discerning the best algorithm for each particular threshold interval, since the one with the highest *subjective quality* value always has the highest *modularity* as well. Following this, Figure 3.5 (left) shows that Louvain's method obtains a better *modularity* score for every threshold value in the interval considered.

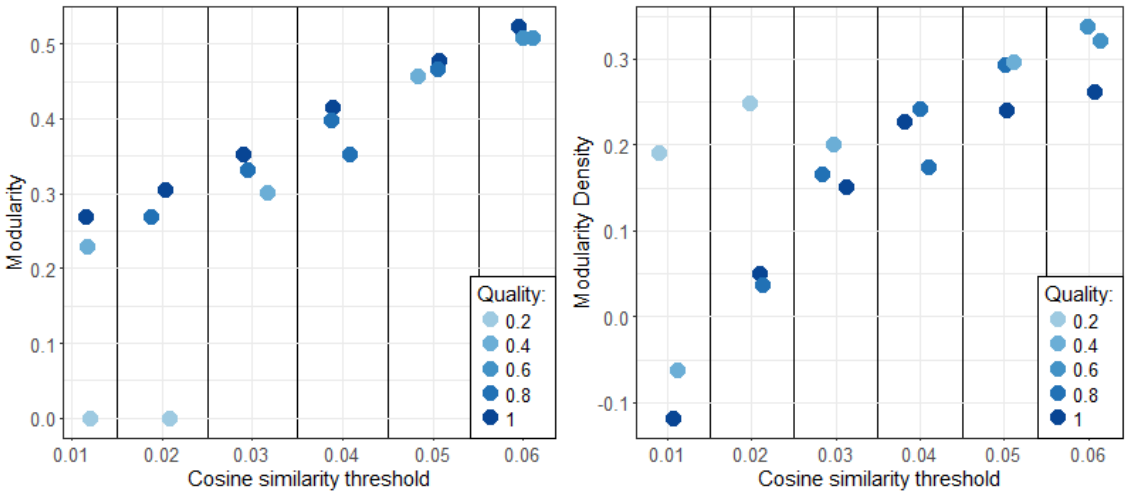

Figure 3.6: *Modularity* (left) and *modularity density* (right) versus threshold, coloured by the *subjective quality*. Some jitter was added to the points to allow for the visualisation of overlapping points.

Using modularity with component penalty

It would be interesting to analyse if the *modularity with component penalty* also awards higher values to the algorithms that have better *subjective quality* for every value in the best threshold interval. Figure 3.7 (left) shows that this is true for the interval of thresholds under analysis. On the right we can see once again that the method that corresponds to these points is Louvain's. This is not very surprising considering that this measure is based on *modularity*.

Threshold	Algorithm	Community Sizes	n	Mod.	Mod. Dens.	Q_p	Quality
0.01	louvain	4, 7, 11, 12, 14, 18	1	0.269	-0.118	0.269	1
0.01	walktrap	6, 6, 22, 32	1	0.230	-0.063	0.230	0.4
0.01	infomap	66	1	0.000	0.191	0.000	0.2
0.02	louvain	4, 7, 7, 7, 10, 13, 18	1	0.305	0.051	0.305	1
0.02	walktrap	3, 4, 6, 10, 20, 23	1	0.270	0.037	0.270	0.8
0.02	infomap	66	1	0.000	0.249	0.000	0.2
0.03	louvain	4, 7, 7, 7, 11, 12, 18	1	0.352	0.153	0.352	1
0.03	walktrap	1, 1, 6, 6, 11, 15, 26	1	0.332	0.167	0.332	0.8
0.03	infomap	3, 6, 7, 11, 39	1	0.300	0.200	0.300	0.4
0.04	louvain	1, 7, 8, 10, 11, 11, 18	2	0.415	0.227	0.115	1
0.04	infomap	1, 3, 3, 7, 7, 9, 10, 26	2	0.399	0.243	0.099	0.8
0.04	walktrap	1, 1, 2, 4, 15, 20, 23	2	0.353	0.174	0.053	0.8
0.05	louvain	1, 7, 9, 10, 10, 13, 16	2	0.478	0.241	0.178	1
0.05	infomap	1, 2, 3, 3, 4, 7, 7, 7, 10, 22	2	0.466	0.294	0.166	0.8
0.05	walktrap	1, 1, 1, 2, 2, 3, 5, 7, 7, 7, 9, 21	2	0.457	0.296	0.157	0.4
0.06	louvain	1, 1, 7, 8, 9, 10, 10, 10, 10	3	0.523	0.261	0.048	1
0.06	infomap	1, 1, 2, 2, 3, 4, 5, 5, 6, 6, 7, 7, 17	3	0.509	0.339	0.033	0.6
0.06	walktrap	1, 1, 2, 3, 4, 6, 6, 7, 8, 8, 20	3	0.508	0.321	0.033	0.6

Table 3.3: Table containing information for each combination of algorithm and threshold evaluated. Ordered by threshold, and in each interval it is ordered by decreasing value of *modularity with component penalty* (Q_p).

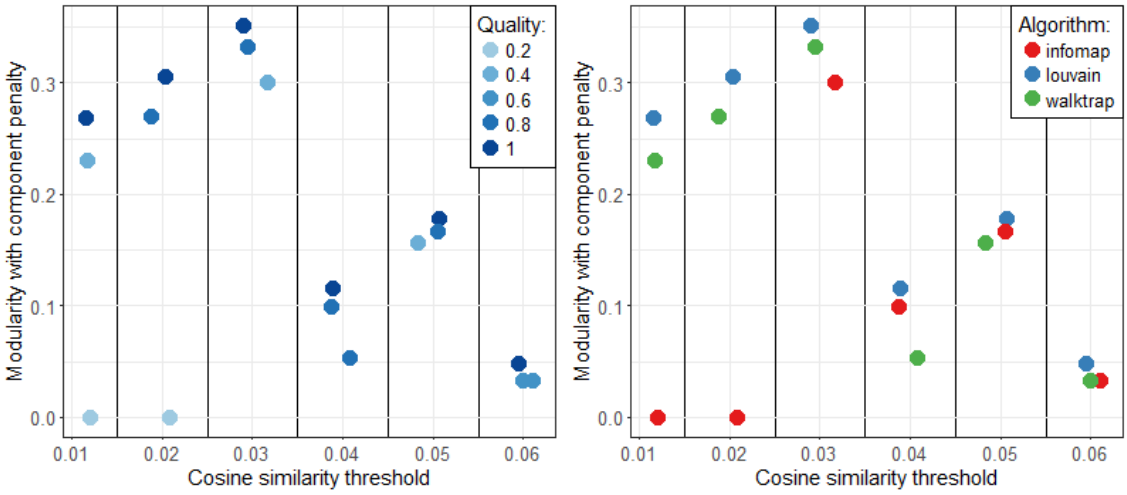

Figure 3.7: *Modularity with component penalty* versus threshold, coloured by the *subjective quality* (left) and by the *algorithm* (right). Some jitter was added to the points to allow for the visualisation of overlapping points.

3.12 Conclusion

The first objective in this chapter was to identify a value for the cosine similarity threshold that would lead to a good representation of the network and a good outcome of the chosen community detection algorithm. In this regard, we concluded that the *modularity* was not an adequate indicator for this task, possibly because different values of the similarity threshold led to different networks and *modularity* is best suited for evaluation of different algorithms on the same network.

I created a *subjective quality* measure based on my evaluation of the partitions of the network obtained for each combination of similarity threshold and community detection algorithm. I also introduced a modified version of the *modularity* referred to as *modularity with component penalty* (Q_p). The penalisation term is represented by the logarithm of the number of components in the network, which is a consequence of increasing the similarity threshold value. I then used the *subjective quality* measure to do a grid search to identify a suitable value for the weight of the penalty. Q_p may not be adequate to find an optimum value, so I was only able to determine an interval of reasonable values. Based on this, I considered the interval of adequate thresholds to be between 0.01 and 0.06 (inclusive).

The second objective in this chapter was to identify the best community detection algorithm for this network. Since I could not select an optimum threshold value, I tested the algorithms for each value of the threshold inside the previously determined interval. By considering the threshold values individually, both *modularity* and *modularity density* are appropriate indicators for this task since the network we are comparing them on is the same. Using the *subjective quality* attribute, I determined *modularity density* to be an inadequate measure to compare the results and used the *modularity* score instead. Analysing the three algorithms *Walktrap*, *Louvain* and *Infomap* for each value of the threshold in the interval I considered good, I concluded that *Louvain*'s method achieved a higher *modularity* score in all of them. I then used the *modularity with component penalty* and obtained the same results, which is not surprising since it is based on *modularity*. As such,

I concluded that Louvain's method is the most adequate algorithm for this study.

In Chapter 4 I explore the data used for this study and for the network used in this chapter. I apply one of the thresholds and community detection methods to the network and analyse it further. Based on the results obtained here, I chose one of the values in the interval of good thresholds, 0.06, and Louvain's method to detect communities. I analyse the affinity groups obtained with these.

Chapter 4

Analysis of the Affinity Groups of FEP Researchers

4.1 Motivation

Every organisation benefits from learning more about its own structure and networks can provide insight into this. The School of Economics and Management of University of Porto (FEP), as an organisation, could benefit from this as well. This was one of the motivations for carrying out this study. In particular, we can analyse the communities of researchers identified to attain a better understanding of the most prominent research trends at the Faculty. This in turn can help to define suitable research groups for future projects. Researchers themselves can use this information to look for and identify other researchers that have carried out similar or complementary work.

The objective of this chapter is to determine if the methods used in Chapter 3 are able to uncover a meaningful structure in the form of a network.

4.2 Organisation of this chapter

In Section 4.3 I describe the original data and the pre-processing involved. Section 4.4 contains a general overview of the network generated from the data and some basic statistics for both network and researchers. In Section 4.5 I analyse the groups identified by applying a community detection algorithm to the network of researchers, and compare those groups to the scientific groups that the Faculty has already defined.

4.3 Data

4.3.1 Original data and pre-processing

The data for this project was provided by the *Authenticus* team (Authenticus, 2017) and spans the period from January 1972 to December 2016. I refer to this dataset as the `Authenticus` dataset. The data contains the titles of the publications of authors that are affiliated with FEP, which amounts to 14 998 authors and 28 238 publications. Many of these authors may not be affiliated with FEP, but they are in the database because they co-authored articles with researchers that were. Using the information in the website of FEP (FEP, 2017), I created a new dataset with the names of the authors that are in the website, listed as either *active* or *inactive*, and the scientific group they belong to, which were retrieved from the website of FEP as well.

There are 5 scientific groups in FEP that researchers can be a part of, exclusively: *Economics*, *Management*, *Maths and Information Science* (referred to as *Maths and InfSci* from this point forward), *Social Sciences* and *Law*. In order to get these groups I simply used the search feature and collected the names of all researchers that belonged to one of the five groups. I refer to this dataset that contains the names and scientific groups of FEP researchers as the `researchers` dataset.

To find the entries in the `Authenticus` dataset that corresponded to authors from the `researchers` dataset, I did the following: first, for each author in the `researchers`

dataset, find one entry in the `Authenticus` dataset that has the same author's name. Then, from this entry extract the researcher's ID. After collecting all the IDs, I went through the `Authenticus` dataset again and retrieved all the entries that contained one of those IDs.

In order to do this I had to clean the data first. In both datasets, the names of the authors were changed to lower case and special characters were replaced with equivalent characters. For example, "ç" was changed to "c" and "é" to "e". Some names in the `researchers` dataset also had to be slightly altered in order to match the names found in the `Authenticus` dataset. For example, the same researcher could be listed as *"Pedro de Sousa"* in one dataset and *"Pedro Sousa"* in the other. This was done in order to avoid missing entries because the names were spelled slightly different. I chose to extract the entries based on the researchers' IDs because there could potentially be some variations in the names of the authors in the `Authenticus` dataset. By using the IDs I only needed to find one entry that matched the name on the `researchers` dataset, and then I could use that ID to find the remaining entries of that author regardless of how their name was spelled. After cleaning the data and extracting the relevant entries, I rearranged the data to match the format required by Affinity Miner.

After extracting only the authors from FEP, the dataset has 102 authors and a total of 1201 publications. I have verified that the dataset that was prepared by the Authenticus team and that I have used here does not include all of the researchers' publications that are available at the Authenticus website, and it may also include articles that are listed as awaiting validation in the website. However, this aspect is relatively insignificant for the analysis carried out here.

The distribution of researchers per scientific group can be found in Figure 4.1. *Economics* is the group with most researchers, with a total of 46. Next comes *Management*, with 30, followed by *Maths and InfSci* with 21. The other two groups, *Social Sciences* and *Law*, are very small with only 3 and 2 authors, respectively. Figure 4.2 shows the number of papers per scientific group. Despite being only the 3rd biggest group, *Maths and InfSci* has the most publications, more than doubling the number of articles in the

Economics and *Management* groups. The *Social Sciences* group has a considerable number of publications when we take into account that it only has 3 authors. The *Law* group has the least number of articles, with just 3. It should be noted that the number of publications in the figure does not coincide with the total number of papers in the dataset. This is because some articles may be counted twice in this figure when they were co-authored by researchers from different groups.

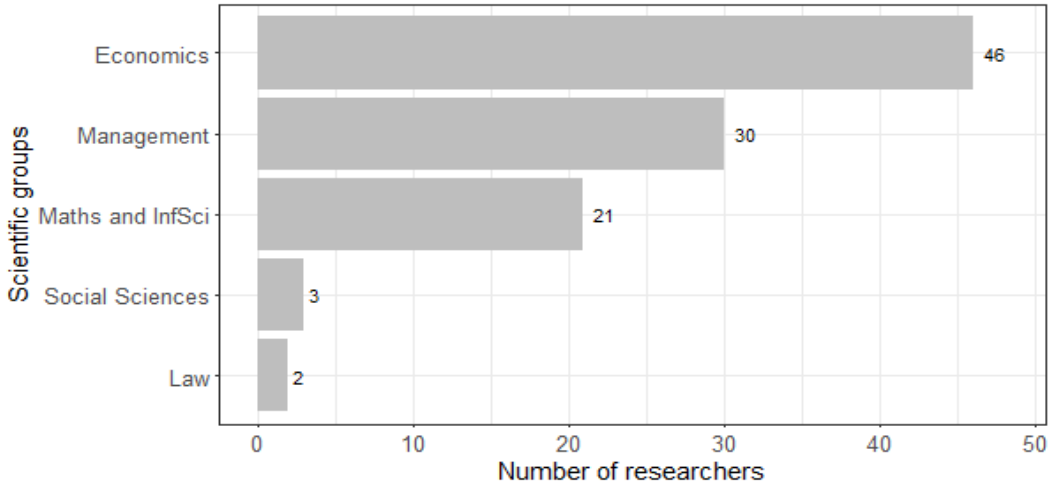

Figure 4.1: Number of researchers per scientific group.

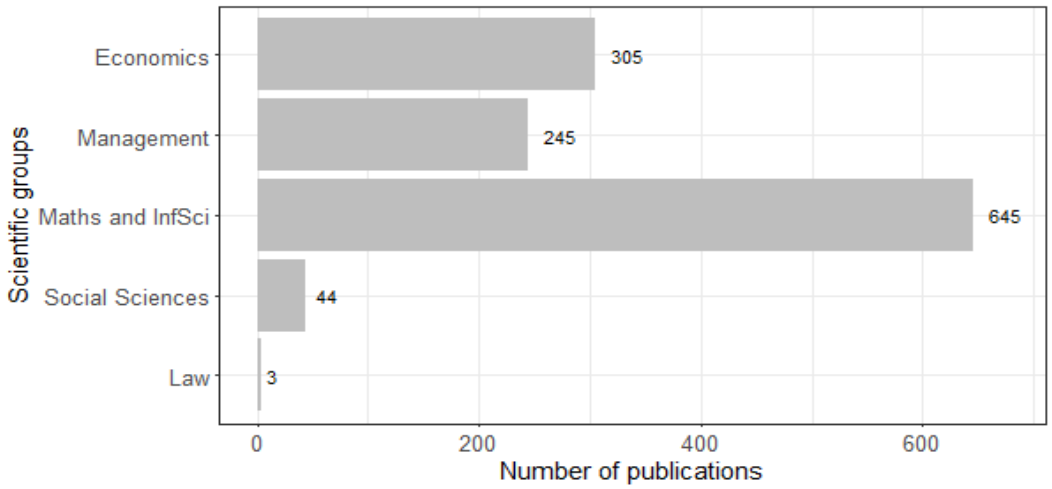

Figure 4.2: Number of papers per scientific group.

4.3.2 Minimum number of publications required

Using all researchers to generate the network may not be adequate because many of them do not have a sufficient amount of articles published. This could distort the graph generated, leading to authors with little to no similarity to any others. These authors would then be disconnected from the rest of the network or at least be in a community composed of the author alone. Besides, the number of titles that Affinity Miner can use to generate keywords for these researchers is very limited, and hence these may not accurately characterise them. With this in mind, I looked at the number of authors in each group and separated them according to the number of articles they had written, which can be seen in Figure 4.3. Both *Economics* and *Management* groups have a high amount of authors with only 1 article published. I chose to analyse only authors with at least 5 publications based on the reasons discussed earlier.

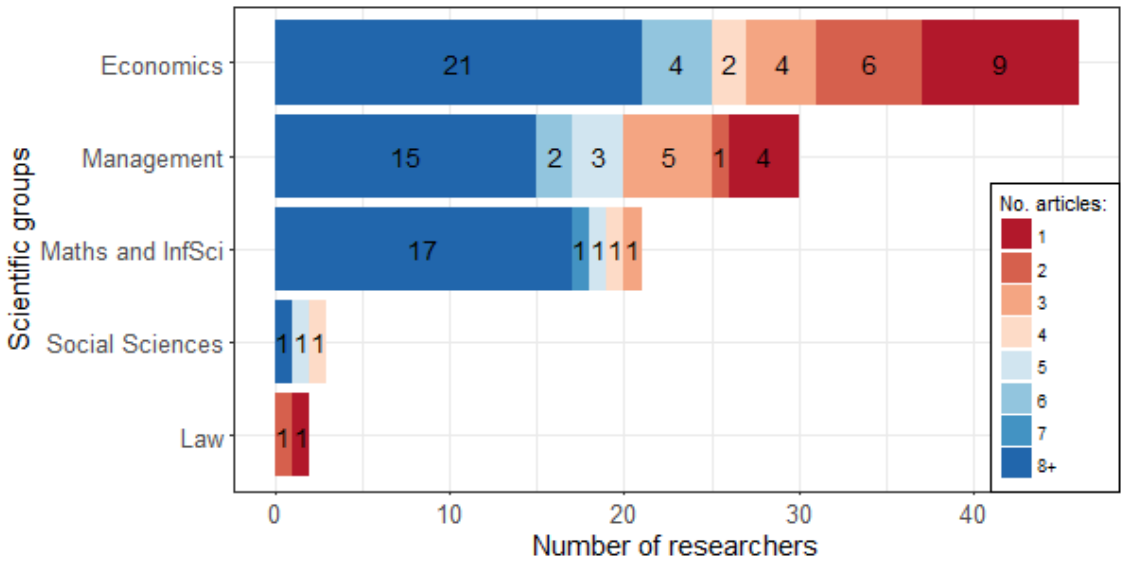

Figure 4.3: Number of researchers per scientific group, with an indication of the number of articles published.

4.3.3 Data used in this study

By only accepting authors with a minimum of 5 publications, the dataset used for this study ended up with 66 researchers with 1149 publications. The new distribution of papers and researchers per scientific group can be found in Figures 4.4 and 4.5. The *Economics* group lost nearly half of its researchers, but only a small portion of its articles. The *Maths and InfSci* group, despite having the largest number of publications, only lost 2 researchers and 7 publications. The *Law* group disappeared since none of its authors had at least 5 publications. As can be seen in Figure 4.3, the researchers in this group only had 1 and 2 articles each.

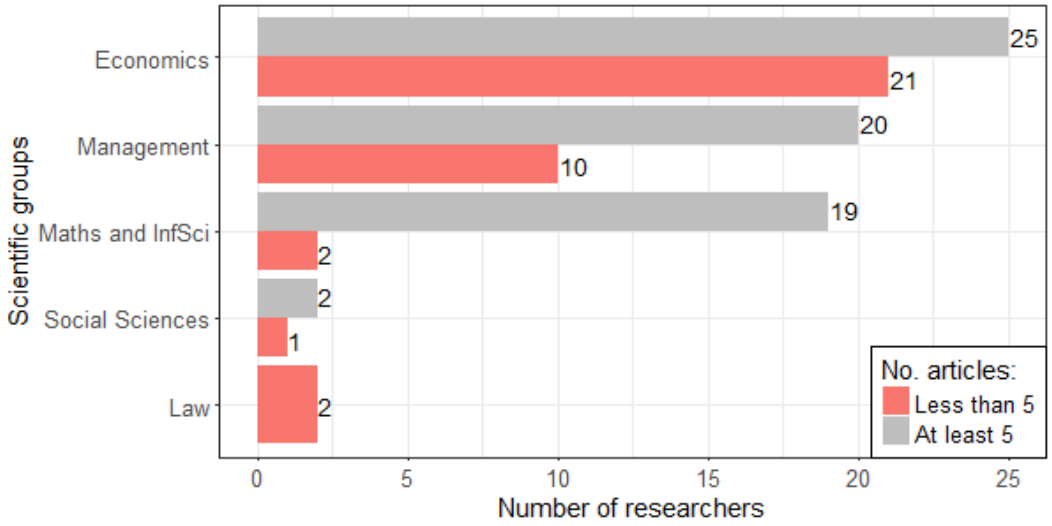

Figure 4.4: Number of researchers per scientific group, separated into two groups: (1) those that have at least 5 publications and (2) those that have less.

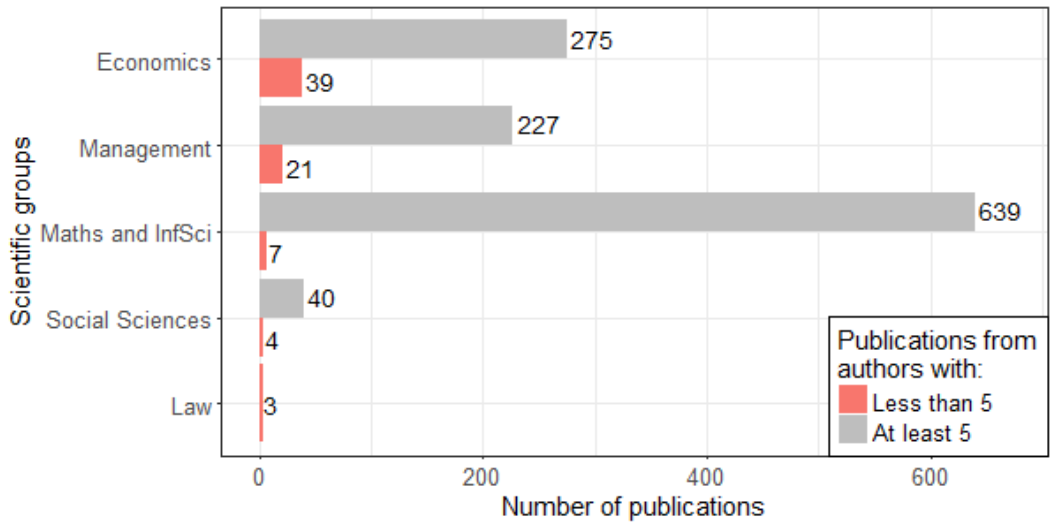

Figure 4.5: Number of papers per scientific group, separated into two groups: (1) those written by authors that have at least 5 publications and (2) those written by authors that have less. Note that papers can be counted more than once, if they belong to authors in different groups or if one author has more than 5 publications and the other less than 5.

4.4 Network of researchers

4.4.1 Description of the network

The network was obtained by creating a document for each researcher containing the titles from their publications. These were then converted into a document-term matrix and the cosine similarity was calculated for each pair of documents. A minimum threshold was then applied to the similarity in order to reduce the number of links in the network. These steps are described in more detail in Sections 3.5 to 3.7.

The threshold chosen for this network was 0.06, a value selected based on the results from Chapter 3. I chose this value because the resulting community sizes were very similar, but any other value between 0.01 and 0.06 could have been chosen instead.

The final network has 66 nodes and 241 links. Each node represents a researcher and the links represent the affinity between him and the researchers they are connected to,

based on the cosine similarity.

Table 4.1 summarises the main characteristics of this network. Not all of the nodes are connected to others, so this network is composed of 3 components, two of which only have one node. The *average degree* is 7, which means that nodes have on average 7 neighbours and 7 links. On average, researchers have some similarity to 7 others. The *average strength* tells us that on average, the *strength* of those 7 links together amounts to 0.97. The *density* shows us that this network is not very dense, only having around 11% of the maximum amount of links this network could have.

The *radius* and *diameter* were calculated using the unweighted version of the network. The two isolated components were also not considered because there is no path from them to any other node, which would make the shortest paths always infinite.

The diameter of the network is 5, so it takes 5 links to connect the two nodes most distant from each other. As for the radius, we need to take every node and the node that is the furthest away from each of them, and consider them as pairs. It takes 3 links to connect the pair that is closest to each other.

Indicator	**Value**
Nodes	66
Links	241
Components	3
Average degree	7.303
Average strength	0.968
Density	0.112
Diameter	5
Radius	3

Table 4.1: Some indicators of the network.

Looking at the distribution of the degree and strength of the researchers in Figure 4.6, it seems that this follows a power-law distribution, with most researchers having a degree and strength around the average or below it, with only a few researchers having the highest values of degree and strength. There seems to be more researchers around the average than at the lower levels of degree and strength.

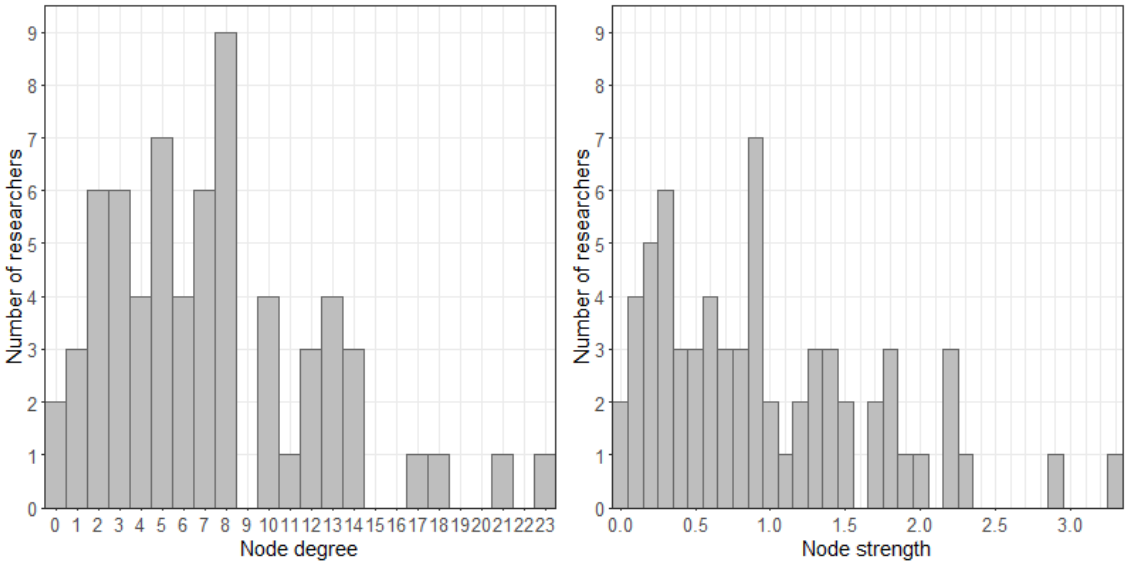

Figure 4.6: Distribution of the degree (left) and strength (right) of the researchers.

4.4.2 Centrality of the researchers

In order to analyse which nodes are the most central in the network, I looked at some indicators: *degree*, *strength* and *betweenness*.The *betweenness* formula contains shortest paths, but since the `igraph` package used to calculate this measure considers lower weights to represent shorter paths, which is the opposite of this network, I used the inverse of the weights of the links.

Some researchers have higher scores in some measures and lower scores in others. However, all of these seem to agree on the 3 most central individuals in the network, as presented in Table 4.2. Interestingly, all of these researchers belong to the scientific group of *Economics*.

Name	**Scientific Group**	**Number of publications**	**Degree**	**Strength**	**Betweenness**
Aurora Amélia Teixeira	Economics	38	23	2.31	302.0
Óscar João Afonso	Economics	49	21	3.26	413.0
António Abílio Brandão	Economics	14	18	2.89	271.0

Table 4.2: Some indicators of centrality for the most central researchers.

In Table 4.3 are the number of researchers from each group in the top ten most central

nodes according to each of these measures. Again, the *Economics* group clearly has the most central nodes.

Scientific Group	Degree	Strength	Betweenness
Economics	8	6	6
Management	1	3	3
Maths and InfSci	1	1	1
Social Sciences	0	0	0

Table 4.3: Number of researchers from each scientific group in the top ten central nodes of each indicator.

4.5 Detection and analysis of affinity groups

Affinity groups

Here I apply Louvain's method for community detection to the network in order to obtain the affinity groups. Table 4.4 shows that the community detection algorithm found 9 affinity groups. The sizes of these groups are balanced, ranging from 7 to 10 authors in each, though the first two affinity groups have only 1 author. These are the two authors that were disconnected from the rest of the network due to the cosine similarity threshold that was applied.

Affinity Group	Size
1	1
2	1
3	10
4	9
5	10
6	10
7	7
8	10
9	8

Table 4.4: Sizes of the affinity groups.

Tables 4.5 and 4.6 contain the researchers in each affinity group and also some of the keywords generated by Affinity Miner to describe each of the groups. The group that

seems to have a better defined set of keywords is group 5, which seems to be mostly about Data Science and Machine Learning. In fact, many of the researchers in this group are part of the Master's Degree in Modelling, Data Analysis and Decision Support Systems. Group 7 also has well defined keywords, which are about optimisation and heuristics. The remaining groups do not have such clear distinctions however.

Affinity Group	**Keywords**	**Members**	**No. Papers**	**Scientific Group**
1	Local Development	Augusto Ernesto Santos Silva	5	Social Sciences
2	Discrete Orbit	Helena Oliveira dos Reis	10	Maths & InfSci
3	Social Responsibility	Carlos Francisco Ferreira Alves	12	Management
	Intellectual Capital	Carlos José Cabral Cardoso	13	Management
	Characteristic Polynomial	Catarina Castelo Branco	20	Management
	Organizational Culture	Francisco Vitorino da Silva Martins	6	Management
	Sustainability Report	João Francisco Alves Ribeiro	10	Management
	Economic Crisis	Luísa Helena Ferreira Pinto	6	Management
	Cross-Cultural Adjustment	Manuel Emílio Castelo Branco	35	Management
	Financial Crisis	Maria Teresa Proença	5	Management
	Portuguese Case	Susana Borges Furtado	15	Maths & InfSci
	Supply Chain Management	José Abílio Oliveira Matos	8	Maths & InfSci
4	Higher Education	Anabela de Jesus Moreira Carneiro	9	Economics
	Maximum Entropy	Elvira Maria de Sousa Silva	11	Economics
	Integral Operator	José Manuel Janeira Varejão	9	Economics
	Portuguese Higher Education	Luís Delfim Moreira dos Santos	6	Economics
	Private Higher Education	Octávio Figueiredo Goncalves	17	Economics
	Spectral Computation	Paulo de Freitas Guimaraes	33	Economics
	Business Cooperation	Paulo Ricardo Tavares Mota	11	Economics
	Economic Growth	Pedro Nuno Lopes Teixeira	35	Social Sciences
	Industrial Location	Paulo José Beleza Vasconcelos	40	Maths & InfSci
5	Data Stream	Adelaide Figueiredo	5	Maths & InfSci
	Knowledge Discovery	Ana Cristina Moreira de Freitas	17	Maths & InfSci
	Data Mining	Carlos Manuel Pinto Soares	114	Maths & InfSci
	Classification Algorithm	Fernanda Figueiredo	23	Maths & InfSci
	Artificial Intelligence	João Manuel Portela da Gama	223	Maths & InfSci
	Change Detection	Jorge Manuel Correia Pereira	27	Maths & InfSci
	Algorithm Selection	José Manuel Soares Oliveira	24	Maths & InfSci
	Machine Learning	Maria Paula Brito	11	Maths & InfSci
	Sensor Data	Pavel Brazdil	69	Maths & InfSci
	Decision Tree	Rui Manuel Rodrigues Leite	9	Maths & InfSci

Table 4.5: Members of affinity groups 1 to 5 and some keywords for each group.

Affinity Group	Keywords	Members	No. Papers	Scientific Group
6	Economic Growth	Ana Paula Africano Silva	6	Economics
	Endogenous Growth	Ana Paula Ferreira Ribeiro	12	Economics
	Costly Investment	Aurora Amélia Castro Teixeira	38	Economics
	Euro Area	Manuel Mota Freitas Martins	10	Economics
	Heart Rate	Maria Isabel Teixeira Soares	37	Economics
	Sea Level	Óscar João Atanazio Afonso	49	Economics
	Wage Inequality	Pedro Rui Mazeda Gil	9	Economics
	Monetary Policy	Rui Henrique Rodrigues Alves	10	Economics
	Welfare Impact	Sandra Maria Tavares Silva	14	Economics
	Ecological Technology	Maria Eduarda Rocha Silva	40	Maths & InfSci
7	Genetic Algorithm	Dalila Martins Fontes	33	Management
	Earlytardy Scheduling	Jorge Miguel Silva Valente	15	Management
	Single Machine Scheduling	José Fernando Gonçalves	30	Management
	Random Key	Maria do Rosário Moreira	12	Management
	Idle Time	Raquel Bastos Moutinho	11	Management
	Tardiness Cost	Rui Alberto Santos Alves	14	Management
	Hybrid Heuristic	Paulo Sérgio Amaral de Sousa	7	Maths & InfSci
8	Cell Network	António Abílio Brandão	14	Economics
	Heteroclinic Network	Hélder Ferreira Vasconcelos	12	Economics
	Welfare Effect	Joana Vaz de Pinho	8	Economics
	Core-Periphery Model	Joana Rita Pinho Resende	11	Economics
	Entry-Exit System	João Oliveira Correia da Silva	14	Economics
	Foreign Direct Investment	Maria Paula Vicente Sarmento	6	Economics
	Natural Gas Market	Rosa Maria Portela Forte	10	Economics
	Asymmetric Collusion	Manuela Alexandrina de Aguiar	13	Maths & InfSci
	Demand Growth	Sofia Dias de Castro Gothen	23	Maths & InfSci
	Investment Decision	Paulo Jorge Ribeiro Pereira	11	Management
9	Cooperation Network	Maria Isabel da Mota Campos	6	Economics
	Virtual Enterprise	Nuno Tiago de Sousa Pereira	9	Economics
	Business Service Network	Pedro José Moreira de Campos	14	Maths & InfSci
	Collaborative Network	Carlos Henrique de Brito	10	Management
	Consumer Market	José Manuel Baptista Mendonca	13	Management
	Supply Management	Maria Catarina de Almeida Roseira	5	Management
	Agent-Based Model	Pedro Manuel Quelhas Brito	12	Management
	Balance Sheet Analysis	Teresa Rocha Fernandes da Silva	5	Management

Table 4.6: Members of affinity groups 6 to 9 and some keywords for each group.

Cross analysis of affinity and scientific groups

Table 4.7 and Figure 4.7 provide a good overview of how the scientific groups populate each affinity group. We can see that affinity groups 3, 7 and 9 are mainly composed of researchers from *Management*. Affinity groups 4, 6 and 8 have authors mostly from

Economics. Group 5 only has researchers from *Maths and InfSci*. It is also worth noting that researchers from *Maths and InfSci* are spread throughout all of the other groups, with the exception of affinity group 1. This comparison allows us to see that the methods used here were able to find a meaningful structure in the network, which corresponds in some degree to the structure created by the scientific groups.

Affinity Group	Size	Economics	Management	Maths and InfSci	Social Sciences
1	1	-	-	-	1
2	1	-	-	1	-
3	10	-	8	2	-
4	9	7	-	1	1
5	10	-	-	10	-
6	10	9	-	1	-
7	7	-	6	1	-
8	10	7	1	2	-
9	8	2	5	1	-

Table 4.7: Number of researchers, total and from each scientific group, that belong to each affinity group.

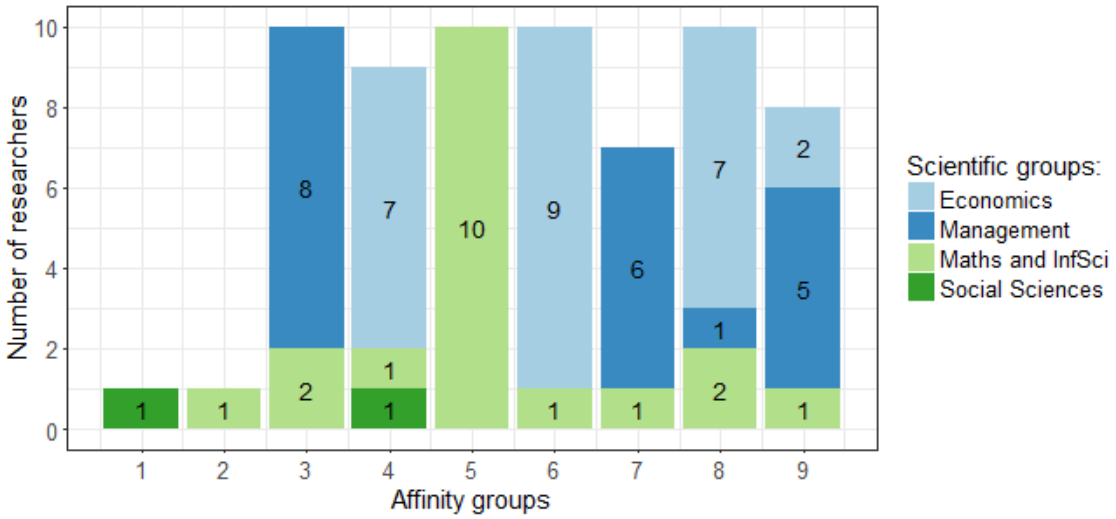

Figure 4.7: Distribution of researchers and scientific groups by affinity group. The colours filling each bar represent the proportion of researchers in that affinity group that belong to the scientific group represented by that colour. The number inside each of those divisions is the number of researchers from that scientific group.

A traditional statistical test cannot be performed due to the low number of researchers for a high number of both affinity and scientific group. An independence test between the

two types of group seems like the most adequate test, but the expected frequencies would not respect Cochran's criteria (Agresti, 2003), even after joining several groups together. If groups were merged further the interpretation of the result of the test would bare little meaning.

4.6 Conclusion

In this chapter I have analysed the network of FEP researchers using a modified version of the Affinity Miner tool. I applied a cosine similarity threshold of 0.06 and used Louvain's method to detect the affinity groups in the network.

I looked at some simple measures of the structure of the network and of the centrality of the individuals. I concluded that the scientific group of *Economics* has the most central individuals, according to the indicators used.

As for the affinity groups, there were 9 groups, 2 of which only had 1 author that was disconnected from the rest of the network. Of the remaining 7, 3 are composed mostly of researchers from *Economics*, 3 have researchers mainly from *Management* and 1 contains researchers only from *Maths and InfSci*. Researchers from this last group are also spread across every group.

In my view, the results of the methods applied in this study are able to find a relevant structure in the network. This is supported by a very positive reaction of some members of this Faculty to the draft of the Working Paper entitled "Analysis of publications of academic staff of FEP and their affinities, with Affinity Miner". These included the Director of the Faculty and some professors.

Chapter 5

Conclusions

5.1 The setting of this work

Networks can be used by many fields of study to gain a better understanding of systems and relations between elements under analysis. It can also be used by companies and institutions for this same purpose.

Affinity Miner is a prototype software that creates networks of researchers based on the text in their publications. In its original development it performed poorly on the community detection task, and as such there was an opportunity to test different methods that could yield better results.

Due to my connection to the School of Economics and Management of the University of Porto (FEP), we have decided to carry out these experiments on data from the Faculty itself, since the results could be relevant for the Faculty as an investigation centre and for its researchers.

The first goal of this study was to represent the researchers of the Faculty and their publications as a network. The second goal was to find a structure in this network that could give us insight into the organisation of the Faculty. To do both of these, I used the data provided by Authenticus (2017). I began by cleaning this data and applied common text pre-processing tasks to generate the network based on the titles of the publications of

the researchers, by creating an adjacency matrix based on the cosine similarity between researchers. I then used community detection algorithms to discover a structure in the network.

5.2 Main results

Introduction of modularity with component penalty

One of the main issues faced was the high number of connections between researchers. In order to simplify the network, I applied a cosine similarity threshold to the links, which would remove the weakest ones. Several values could be used in combination with the community detection algorithms. To test them, I began by using some evaluation measures commonly found in literature, such as the *modularity* and *modularity density*. I have created a *subjective quality* measure according to which I scored each combination, based on what I considered to be adequate sizes for the detected communities. Balanced sizes were favoured, while outcomes with many small communities or a few very large ones did not seem very useful and hence were attributed low scores of *subjective quality*.

Using this measure, I concluded that both modularities were not adequate for this case. This is mainly because they are supposed to be used when comparing different community detection results for the same network. By applying the similarity threshold I am effectively changing the network, so these indicators become less reliable.

I have created an alternative measure based on the penalisation of some configurations. This new measure, which I named *modularity with component penalty*, penalised solutions by the number of researchers that were disconnected from the rest of the network. Using the *subjective quality* attribute and grid search, I discovered an appropriate weight for this penalty. This new measure allowed me to determine an interval of thresholds that yielded satisfactory results from the community detection algorithms, in regards to the sizes of the communities discovered. This interval of thresholds was 0.01 to 0.06.

Analysis of different community detection algorithms

I then focused on the community detection algorithms themselves. I tested 3 algorithms – Walktrap, Louvain and Infomap – for each level of threshold in the interval previously obtained. Since for each level of threshold the network is the same, *modularity* and *modularity density* could be adequate for this study. I analysed the scores the algorithms obtained in these two indicators and *modularity with component penalty*, and compared those scores to the *subjective quality* attribute. I concluded that both the *modularity* and *modularity with component penalty* were more adequate because they favoured combinations that had a high *subjective quality* attribute. According to them the best algorithm to use was Louvain's method, for all levels of threshold tested.

Analysis of the affinity groups at FEP

Having tested these two parameters, I selected one value from the best cosine threshold interval. Since Louvain's method had obtained the best results, I used it for the task of community detection in this pruned network. I studied some basic statistics of both the network and its researchers and found that the group of *Economics* has the most central researchers. There were 9 affinity groups, two of which only had 1 researcher (groups 1 and 2). I looked at the remaining affinity groups and the keywords generated for them, and concluded that there were two groups with clearly defined areas of study: the group of data science, data analysis and machine learning (group 5), and the group of optimisation and heuristics (group 7). The other groups were not so clear.

This was followed by a comparison of the scientific groups defined by the Faculty and the affinity groups discovered by the algorithm, in which I noticed that affinity groups are mostly composed of researchers from one particular scientific group. Affinity groups 4, 6 and 8 are mainly composed of researchers from *Economics*, while groups 3, 7 and 9 have a majority of researchers from *Management*. Group 5 only has researchers from *Maths and InfSci*. It is also interesting that this scientific group has researchers in nearly all of

the affinity groups. This analysis suggests that the methods used in this study were able to find a pertinent structure in the network. Some members of the Faculty also had a positive reaction to the structure discovered.

The results obtained in this study allow us to discover interesting relationships in this group of researchers, and those familiar with the structure of the Faculty may find these to accurately represent their own perception of the real-world network or even give them new insights.

5.3 Future work

This study was the basis for a report titled "Analysis of publications of academic staff of FEP and their affinities, with Affinity Miner". This report introduces the relevance of performing a network analysis of the Faculty and the advantages it can bring to it. Affinity Miner is presented as an easy way to visualise all of this information. This report is currently nearly complete and will be submitted to a FEP Working Paper. Some members of the Faculty have already reacted positively to our draft.

There is still room for future research:

- The pruning method used to simplify the network is rather simple and may diminish the effectiveness of the community detection method. One major consequence observed in this study was the cutting of the network into several components. As such, using other pruning methods that keep the network connected could provide better affinity groups, specially for a higher percentage of links removed.

- The *subjective quality* measure was biased since it was based solely on my opinion. Using a more complex scoring system that takes into account the opinions of several human judges would be less biased.

- Only three algorithms were experimented with in this study. However, there are

others which could have been used instead, and so testing them could provide interesting results.

– Though it was not the aim of this particular study to focus on the generated keywords, these do not seem to be satisfactory in some cases. Their quality should be evaluated and other methods of keyword generation should be tested.

– Finally, one major improvement would be to apply these methods not just to the School of Economics and Management of the University of Porto but to the entire University. This would lead to a vast network, which in turn would lead to new problems to solve. For example, the running time of the algorithms would have to be evaluated. This was not a concern of this study because the network is fairly small, but if we were to include all the of researchers from the University it would most likely affect the performance of the methods used.

Regarding the introduction of the new measure *modularity with component penalty* our plan is to discuss it in a separate paper and submit it to a scientific workshop, conference or journal.

Bibliography

Agresti, A. (2003). *Categorical Data Analysis*. Wiley, second edition.

Authenticus (2017). Authenticus bibliographic database. `https://www.authenticus.pt/`, accessed on 14 January 2017.

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, Vol. 286(5439):509–512.

Barrat, A., Barthélemy, M., Pastor-Satorras, R., and Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 101(11):3747–3752.

Berry, J. W., Hendrickson, B., Laviolette, R. A., and Phillips, C. A. (2011). Tolerating the community detection resolution limit with edge weighting. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, Vol. 83(5):1–9.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*.

Bohlin, L., Edler, D., Lancichinetti, A., and Rosvall, M. (2014). Community Detection and Visualization of Networks with the Map Equation Framework. *Measuring Scholarly Impact*, pages 3–34.

Bonacich, P. (1987). Power and Centrality: A Family of Measures. *American Journal of Sociology*, Vol. 92(5):1170–1182.

Brazdil, P., Trigo, L., Cordeiro, J., Sarmento, R., and Valizadeh, M. (2015). Affinity mining of documents sets via network analysis, keywords and summaries. *Oslo Studies in Language*, Vol. 7(1).

Büttcher, S., Clarke, C., and Cormack, G. V. (2010). *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press.

Chen, M., Nguyen, T., and Szymanski, B. K. (2013). On Measuring the Quality of a Network Community Structure. In *2013 International Conference on Social Computing*, pages 122–127.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Vol. 1695(5):1–9.

Easley, D. and Kleinberg, J. (2010). *Networks, Crowds and Markets: Reasoning about*

a Highly Connected World. Cambridge University Press, New York, United States of America, 1st edition.

Feldman, R. and Sanger, J. (2007). *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

FEP (2017). Faculdade de Economia da Universidade do Porto. `https://sigarra.up.pt/fep/pt/`, accessed on 10 February 2017.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Pnas*, Vol. 104(1):36–41.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, Vol. 1(3):215–239.

Girvan, M. and Newman, M. E. K. (2002). Community structure in social and biological networks. *Proct Natl Acad Sci USA*, Vol. 99(12):7821–7826.

Granovetter, M. S. (1973). The strength of weak ties. *American journal of sociology*, Vol. 78(6):1360–1380.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, Vol. 2(1):193–218.

Iacobucci, D. (1994). Graphs and Matrices. In Wasserman, S. and Faust, K., editors, *Social Network Analysis: Methods and Applications*, chapter Graphs and, pages 92–166. Cambridge University Press, New York.

Kaufman, L. and Rousseeuw, P. J. (2005). *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons, Inc.

Latapy, M. and Pons, P. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, Vol. 10(2):191–218.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of EMNLP*, Vol. 85:404–411.

Newman, M. (2003a). The structure and function of complex networks. *SIAM Review*, Vol. 45(2):167–256.

Newman, M. E. (2001). Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, Vol. 64(1).

Newman, M. E. J. (2003b). Mixing patterns in networks. *Physical Review E*, Vol. 67(2).

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, Vol. 69(2).

Oliveira, M. and Gama, J. (2012). An overview of social network analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 2(2):99–115.

Opsahl, T., Agneessens, F., and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, Vol. 32(3):245–251.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rand, M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, Vol. 66(336):846–850.

Rosvall, M., Axelsson, D., and Bergstrom, C. T. (2009). The map equation. *European Physical Journal: Special Topics*, Vol. 178(1):13–23.

Rosvall, M. and Bergstrom, C. T. (2007). An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 104(18):7327–7331.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 105(4):1118–1123.

Steinhaeuser, K. and Chawla, N. V. (2010). Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, Vol. 31(5):413–421.

Tan, P., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Pearson Addison Wesley.

Travers, J. and Milgram, S. (1969). An Experimental Study of the Small World Problem. *Sociometry*, Vol. 32:425–443.

Trigo, L., Víta, M., Sarmento, R., and Brazdil, P. (2015). Retrieval, Visualization and Validation of Affinities between Documents. In *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2015*.

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, Vol. 58(301):236–244.

Wasserman, S. and Faust, K. (1994). *Social network analysis: methods and applications*, volume 8. Cambridge University Press, Cambridge, UK.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, Vol. 393(6684):440–442.

Annex A

Data

A.1 `researchers` dataset

The data in this Section was retrieved from the website of FEP (FEP, 2017) in February 2017. This information is publicly available. The `name` column contains the names of the researchers and the `group` column has the scientific groups they belong to.

	name	**group**
1	abel luis costa fernandes	economics
2	alvaro fernando santos almeida	economics
3	alvaro pinto coelho de aguiar	economics
4	ana paula africano sousa silva	economics
5	ana paula dias delgado	economics
6	ana paula ferreira ribeiro	economics
7	ana teresa cunha de pinho tavares lehmann	economics
8	anabela de jesus moreira carneiro	economics
9	antonio abilio garrido da cunha brandao	economics
10	antonio carlos fernandes teixeira	economics
11	argentino conceicao da silva pessoa	economics
12	armindo manuel da silva carvalho	economics
13	aurora amelia castro teixeira	economics
14	carlos jose gomes pimenta	economics
15	elisa maria da costa guimaraes ferreira	economics
16	elvira maria de sousa silva	economics
17	fernando teixeira dos santos	economics
18	filipe macedo pinto grilo	economics

19	francisco antonio fernandes barros castro	economics
20	francisco luis ferreira nunes pereira	economics
21	helder ferreira vasconcelos	economics
22	helder manuel valente da silva	economics
23	joana patricia neves vaz de pinho	economics
24	joana rita pinho resende	economics
25	joao armando lobo de sousa couto	economics
26	joao manuel de matos loureiro	economics
27	joao oliveira correia da silva	economics
28	jose da silva costa	economics
29	jose manuel janeira varejao	economics
30	jose manuel peres jorge	economics
31	luis delfim pereira moreira dos santos	economics
32	manuel antonio mota freitas martins	economics
33	manuel duarte da silva rocha	economics
34	manuel jose mendes de oliveira	economics
35	manuel luis guimaraes da costa	economics
36	maria clementina pereira nunes teixeira dos santos	economics
37	maria cristina guimaraes guerreiro chaves	economics
38	maria da conceicao pereira ramos	economics
39	maria do pilar esteves gonzalez	economics
40	maria isabel goncalves da mota campos	economics
41	maria isabel rebelo teixeira soares	economics
42	maria manuel de penha dinis correia de pinho	economics
43	maria manuela de castro e silva ferreira	economics
44	maria margarida fernandes ruivo	economics
45	maria margarida malheiro queiroz de mello	economics
46	maria paula vicente sarmento	economics
47	mario alencoao brigido da graca moura	economics
48	mario alexandre patricio martins da silva	economics
49	mario rui sousa moreira da silva	economics
50	miguel jose ferros pimentel reis da fonseca	economics
51	mustafa alper cenesiz	economics
52	natercia silva fortuna	economics
53	nuno alexandre meneses bastos moutinho	economics
54	nuno tiago bandeira de sousa pereira	economics
55	octavio manuel dias de figueiredo goncalves	economics
56	oscar joao atanazio afonso	economics
57	paulo de freitas guimaraes	economics
58	paulo ricardo tavares mota	economics
59	pedro cosme da costa vieira	economics
60	pedro rui mazeda gil	economics
61	rosa maria correia fernandes portela forte	economics

62	rui henrique ribeiro rodrigues alves	economics
63	sandra maria tavares silva	economics
64	susana maria sampaio pacheco pereira de oliveira	economics
65	suzana margarida dias dos santos cavaco	economics
66	vitor manuel da costa carvalho	economics
67	elda oliveira marques	law
68	jose augusto mendes almeida	law
69	maria natalia faria dos santos goncalves	law
70	mariana fontes da costa	law
71	miguel duarte goncalves bras da cunha	law
72	noel barbosa leao pereira gomes	law
73	nuno francisco de sa e melo de castro marques	law
74	alipio jose silva da torre	management
75	amelia maria pinto da cunha brandao	management
76	ana paula de sousa freitas madureira serra	management
77	ana paula marques	management
78	angela maria duarte gago	management
79	antonio de melo da costa cerqueira	management
80	beatriz da graca luz casais	management
81	carlos francisco ferreira alves	management
82	carlos henrique figueiredo e melo de brito	management
83	carlos jose cabral cardoso	management
84	catarina judite morais delgado castelo branco	management
85	claudia alexandra goncalves correia ribeiro	management
86	dalila benedita machado martins fontes	management
87	eduardo andre da silva oliveira	management
88	elisio fernando moreira brandao	management
89	fabiane valeria de oliveira bastos valente	management
90	fernando da costa lima	management
91	fernando fabian cortinas	management
92	francisco vitorino da silva martins	management
93	graca maria azevedo maciel amaro	management
94	hortensia maria da silva gouveia barandas	management
95	isabel margarida paiva de sousa	management
96	isabel maria da silva goncalves leitao da cunha	management
97	joao francisco da silva alves ribeiro	management
98	joao manuel de frias viegas proenca	management
99	joao pedro figueiredo ferreira de carvalho oliveira	management
100	joaquim manuel faria barreiros	management
101	jorge bento ribeiro barbosa farinha	management
102	jorge miguel silva valente	management
103	jose antonio cardoso moreira	management
104	jose fernando goncalves	management

105	jose manuel de araujo baptista mendonca	management
106	jose pedro coelho rodrigues	management
107	julio fernando seara sequeira da mota lobao	management
108	julio manuel dos santos martins	management
109	krinstian philipsen	management
110	leandro manuel ferreira de oliveira	management
111	luis filipe campos dias de castro reis	management
112	luis miguel rodrigues miranda da rocha	management
113	luisa claudia lopes agante	management
114	luisa helena ferreira pinto	management
115	manuel antonio fernandes da graca	management
116	manuel emilio mota de almeida castelo branco	management
117	manuel jose rodrigues da cunha pereira	management
118	manuel marques da costa pinho	management
119	maria catarina de almeida roseira	management
120	maria do rosario mota de oliveira alves moreira	management
121	maria helena goncalves martins	management
122	maria teresa teixeira de carvalho marinho bianchi	management
123	maria teresa vieira campos proenca	management
124	miguel augusto gomes sousa	management
125	nuno ricardo de oliveira moreira	management
126	pallassana krishnan kannan	management
127	patricia andrea bastos teixeira lopes couto viana	management
128	paulo jorge marques de oliveira ribeiro pereira	management
129	pedro manuel dos santos quelhas taumaturgo de brito	management
130	raquel filipa do amaral chambre de meneses soares bastos moutinho	management
131	renata blanc esteves bento de melo	management
132	ricardo miguel araujo cardoso valente	management
133	robert elliott spencer	management
134	rui alberto ferreira santos alves	management
135	rui manuel pinto couto viana	management
136	samuel cruz alves pereira	management
137	teresa maria rocha fernandes da silva	management
138	torben damgaard	management
139	vasco jose de castro viana	management
140	vitor manuel da silva macedo	management
141	adelaide maria de sousa figueiredo	maths
142	alexandra patricia horta ramos	maths
143	ana cristina gomes monteiro moreira de freitas	maths
144	carlos manuel milheiro de oliveira pinto soares	maths
145	fernanda otilia de sousa figueiredo	maths
146	helena maria monteiro moreira oliveira dos reis	maths
147	joao manuel portela da gama	maths

148	jorge manuel correia pereira	maths
149	jose abilio oliveira matos	maths
150	jose manuel soares oliveira	maths
151	manuela alexandrina david de aguiar	maths
152	maria eduarda rocha pinto augusto silva	maths
153	maria paula de pinho de brito duarte silva	maths
154	paulo joao figueiredo cabral teles	maths
155	paulo jose abreu beleza vasconcelos	maths
156	paulo sergio amaral de sousa	maths
157	pavel brazdil	maths
158	pedro jose ramos moreira de campos	maths
159	rui manuel santos rodrigues leite	maths
160	sofia balbina santos dias de castro gothen	maths
161	susana margarida figueiredo de sousa borges furtado	maths
162	vitor manuel martins de matos	maths
163	antonio maria braga de macedo de castro henriques	social
164	augusto ernesto santos silva	social
165	diogo campos monteiro de melo lourenco	social
166	helena maria de azevedo coelho dos santos	social
167	pedro nuno de freitas lopes teixeira	social
168	sofia alexandra soares de miranda ferreira cruz	social

Table A.1: `researchers` dataset after removing special characters.

A.2 Format of the `Authenticus` dataset

In this section I present the main columns of the dataset provided by Authenticus (Authenticus, 2017). Each entry is a publication, and it has up to two researchers in it. If a publication has more than 2 authors, the publication will have one entry for each pair of researchers. The main fields are:

id Entry ID.

rid1 ID of the first researcher in the entry.

rid2 ID of the second researcher in the entry.

rid1_researcher_name Name of the first researcher.

rid2_researcher_name Name of the second researcher.

publication_title Title of the publication.

A.3 Format of the input data of Affinity Miner

At the current stage in development, the data introduced in Affinity Miner must follow a specific format. It should contain a header and it must have 8 columns, of which only four are currently used:

ID The ID of the researcher;

Name The name of the researcher;

Corpus The corpus to be used;

Department The department/group the researcher belongs to.

The remaining columns need to be in the right places, but since they have no use at the moment they can be filled with anything. The rows can either be one line per researcher, where the entire corpus is placed in that row, or it can be one article per row, which would translate to multiple rows per researcher. An example of how the data should look like can be found in Table A.2.

rid	name	col3	title	col5	group	col7	col8
1	John	0	Title1	0	Group1	0	0
2	Mary	0	Title2	0	Group2	0	0
10	Jack	0	Title1	0	Group8	0	0
1	John	0	Title4	0	Group1	0	0

Table A.2: Example of the format of the data to input in Affinity Miner.

Annex B

Visualising affinity and scientific groups

B.1 Affinity groups

Figure B.1 shows the entire network. Figures B.2 to B.8 show the network but highlighting the nodes and links belonging to each affinity group separately. Affinity groups 1 and 2 are not displayed because they are not connected to any nodes, and thus are not represented in the graph.

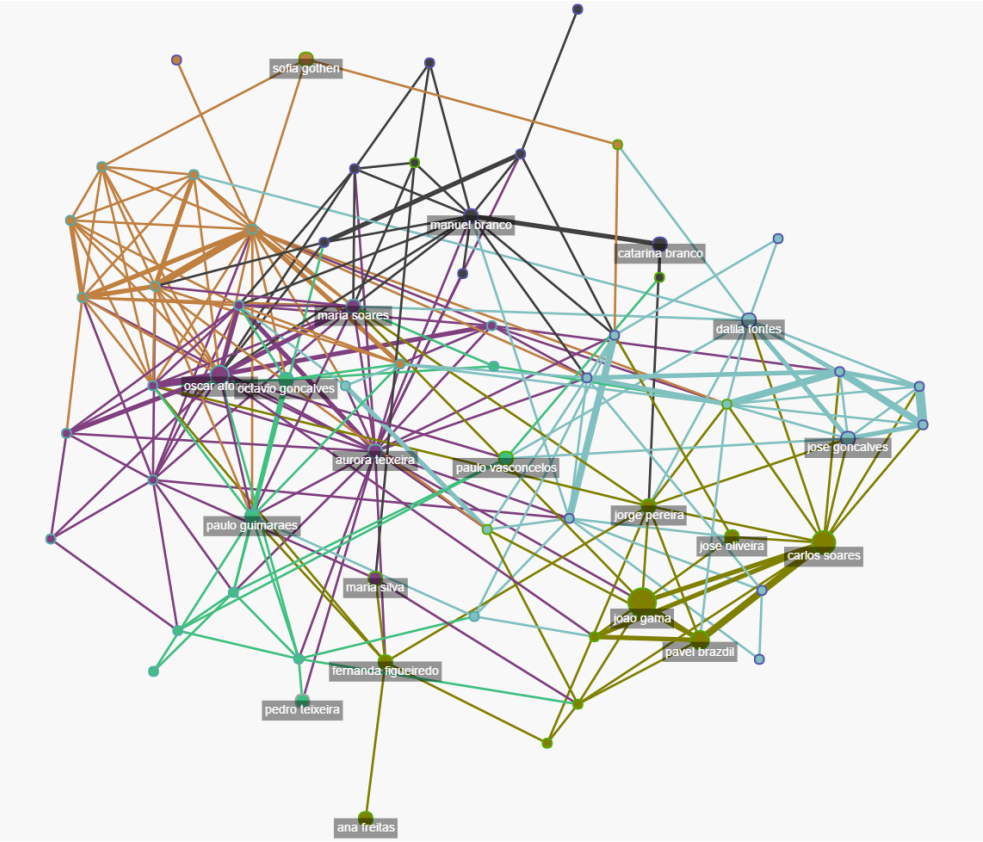

Figure B.1: Full network.

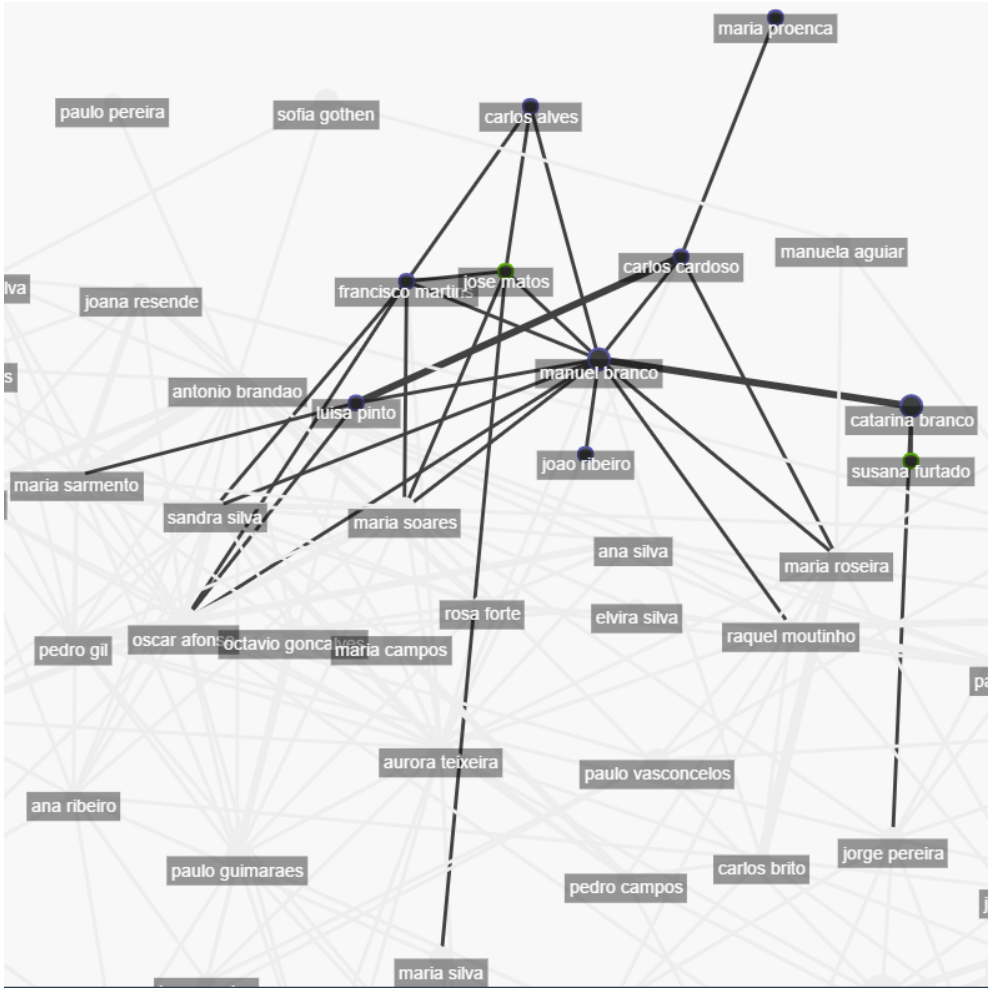

Figure B.2: Affinity group 3.

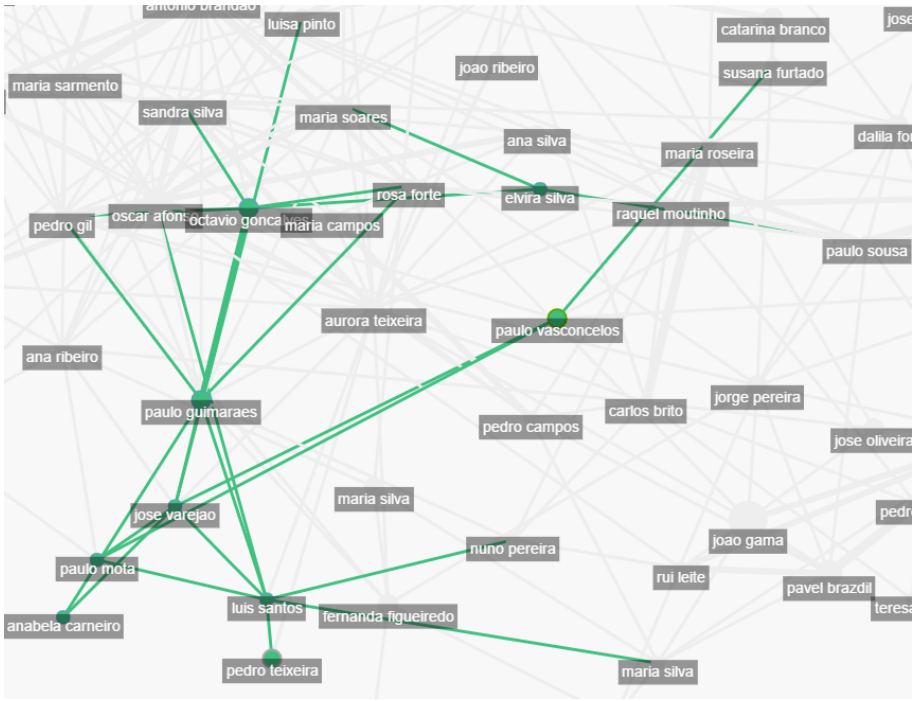

Figure B.3: Affinity group 4.

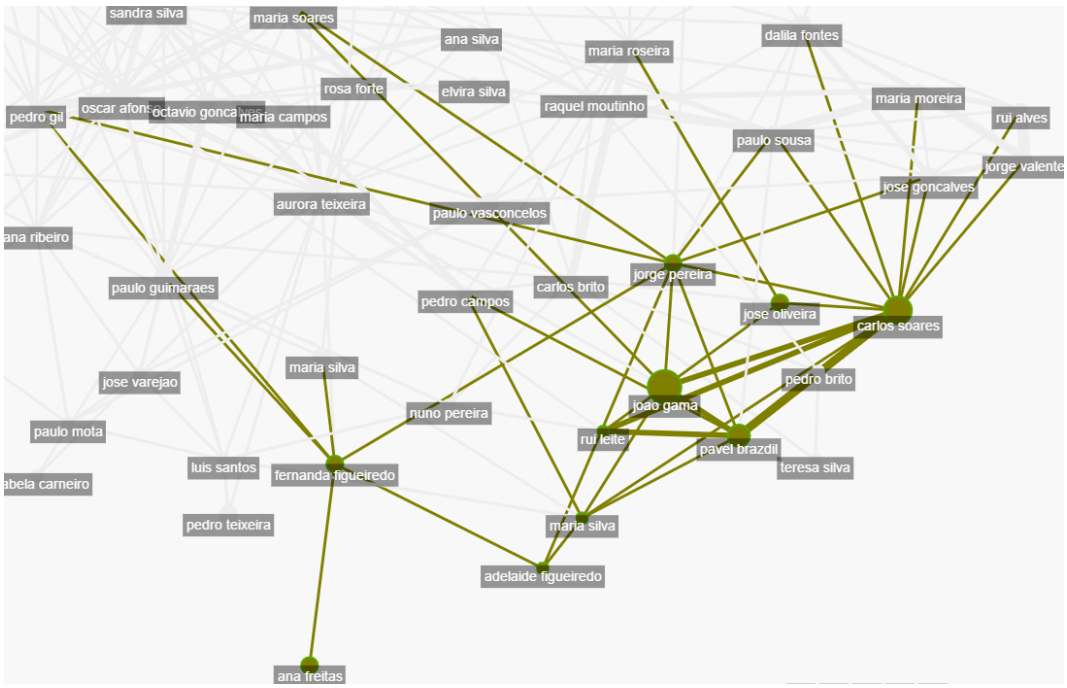

Figure B.4: Affinity group 5.

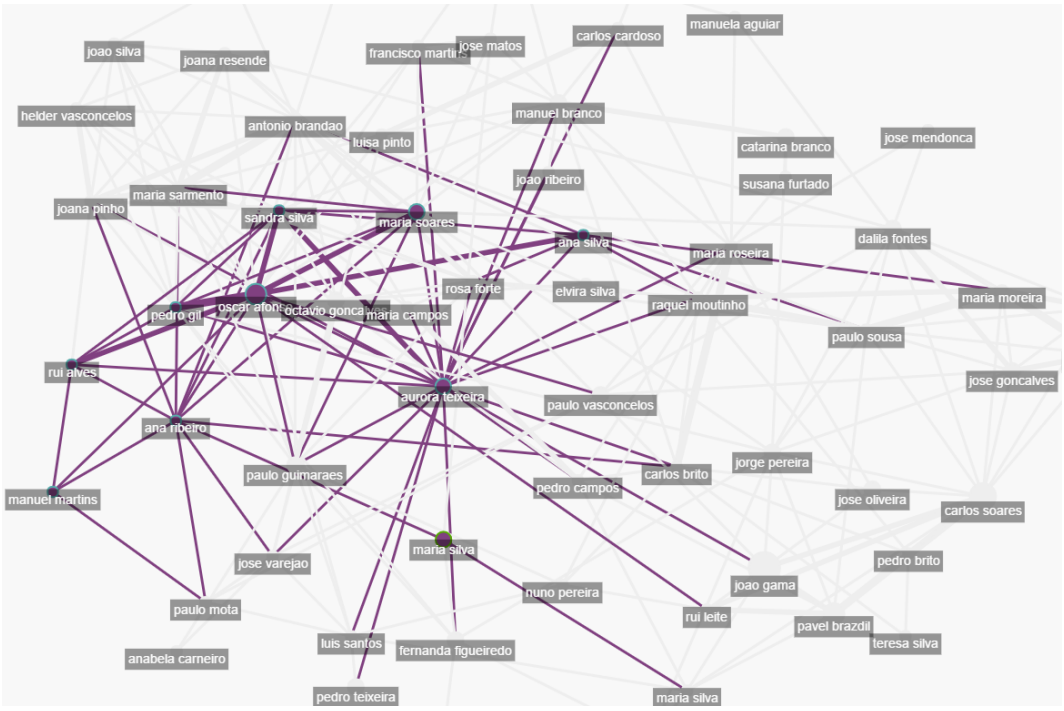

Figure B.5: Affinity group 6.

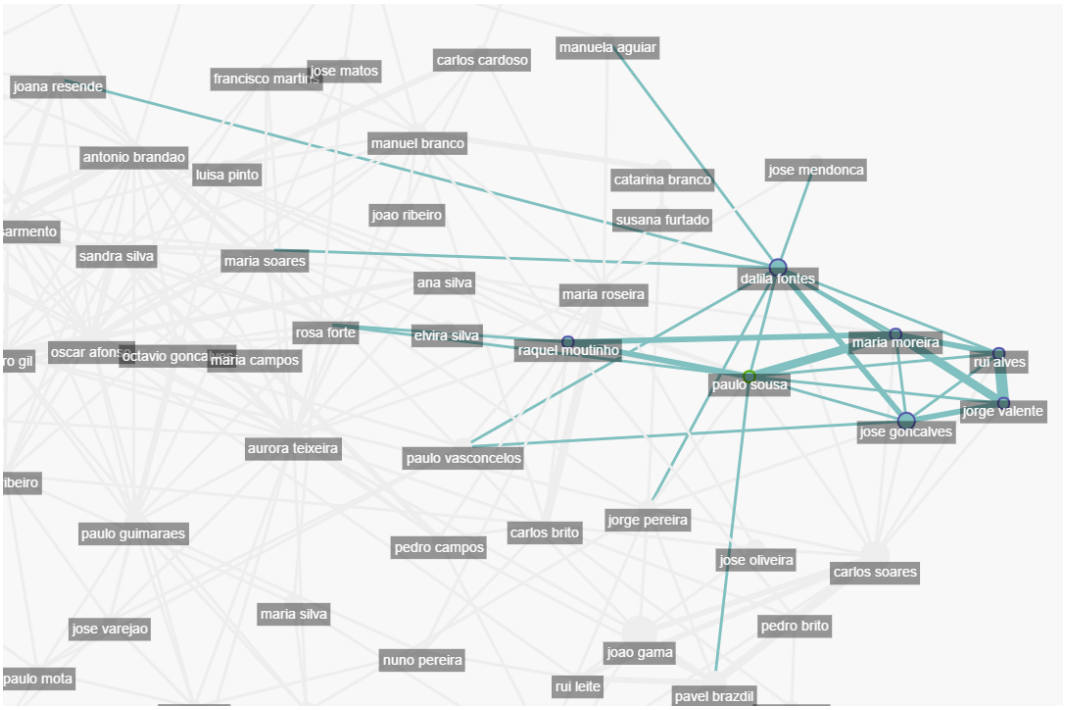

Figure B.6: Affinity group 7.

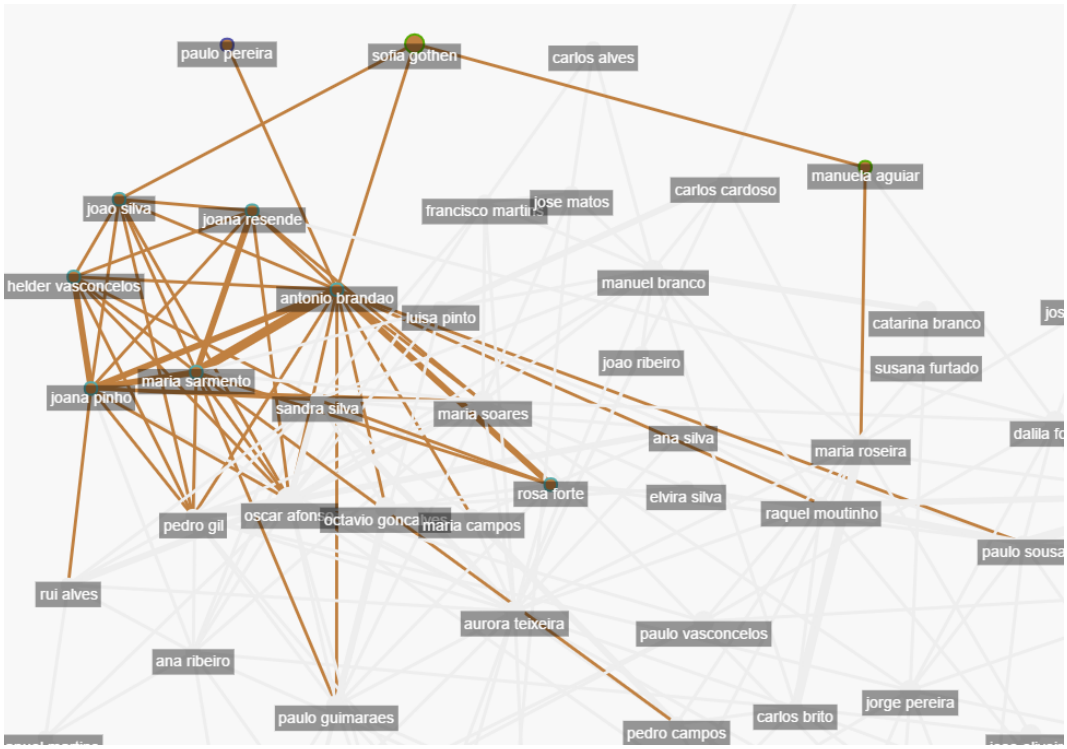

Figure B.7: Affinity group 8.

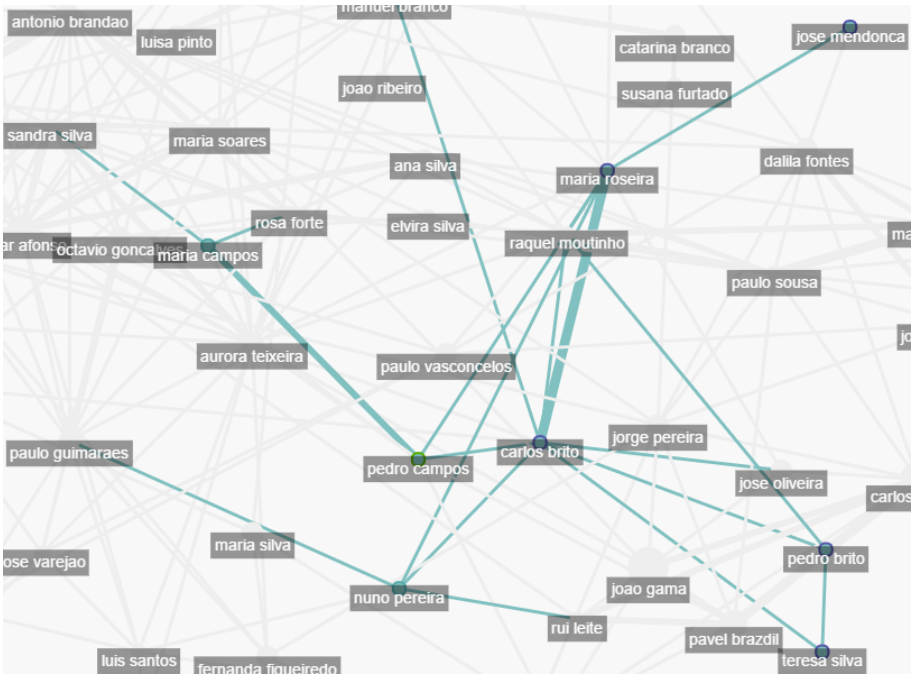

Figure B.8: Affinity group 9.

B.2 Scientific groups

In Figures B.9, B.10 and B.11 are the networks generated for each scientific group. The *Social Sciences* group is not here because its researchers are not connected to each other, and as such Affinity Miner does not plot them.

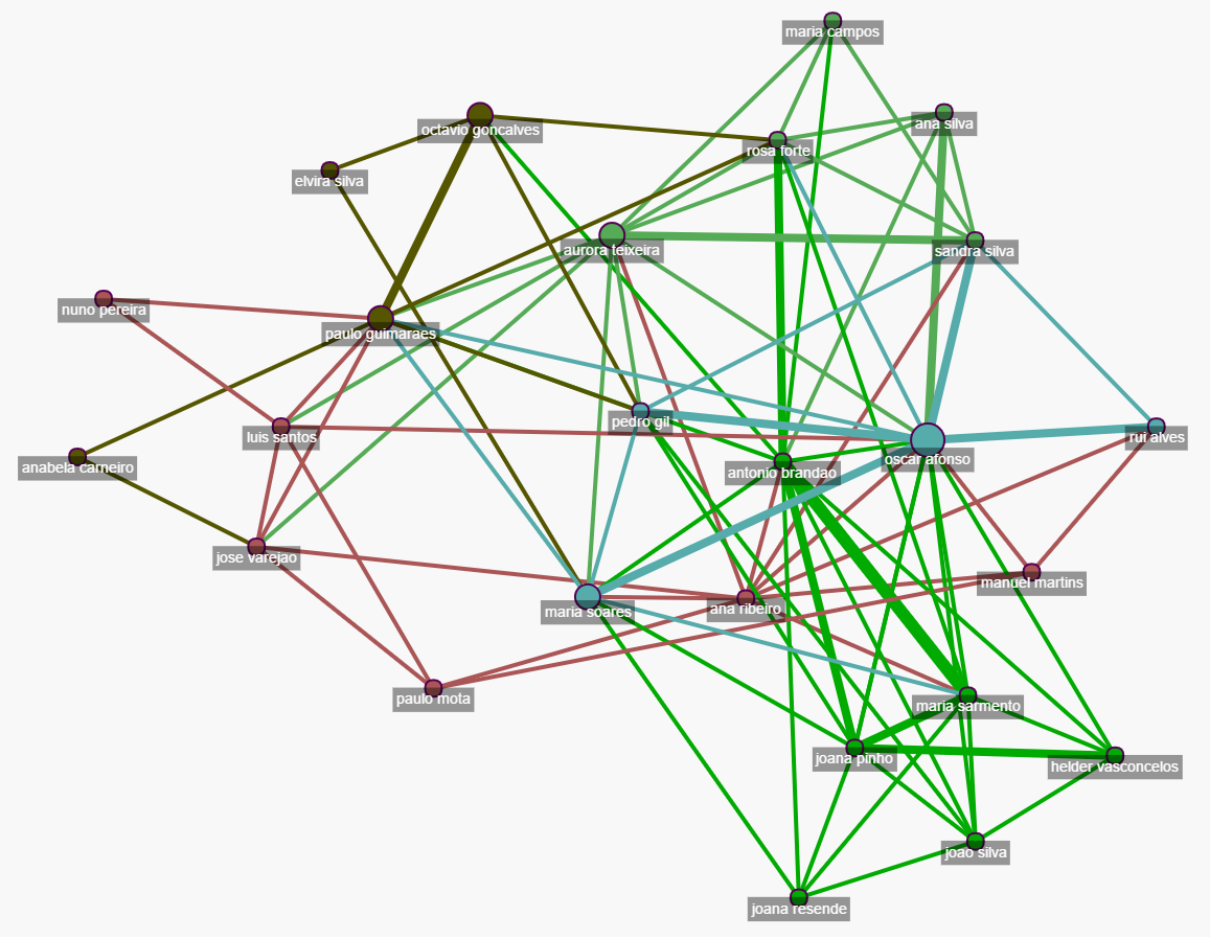

Figure B.9: Network and affinity groups for *Economics*.

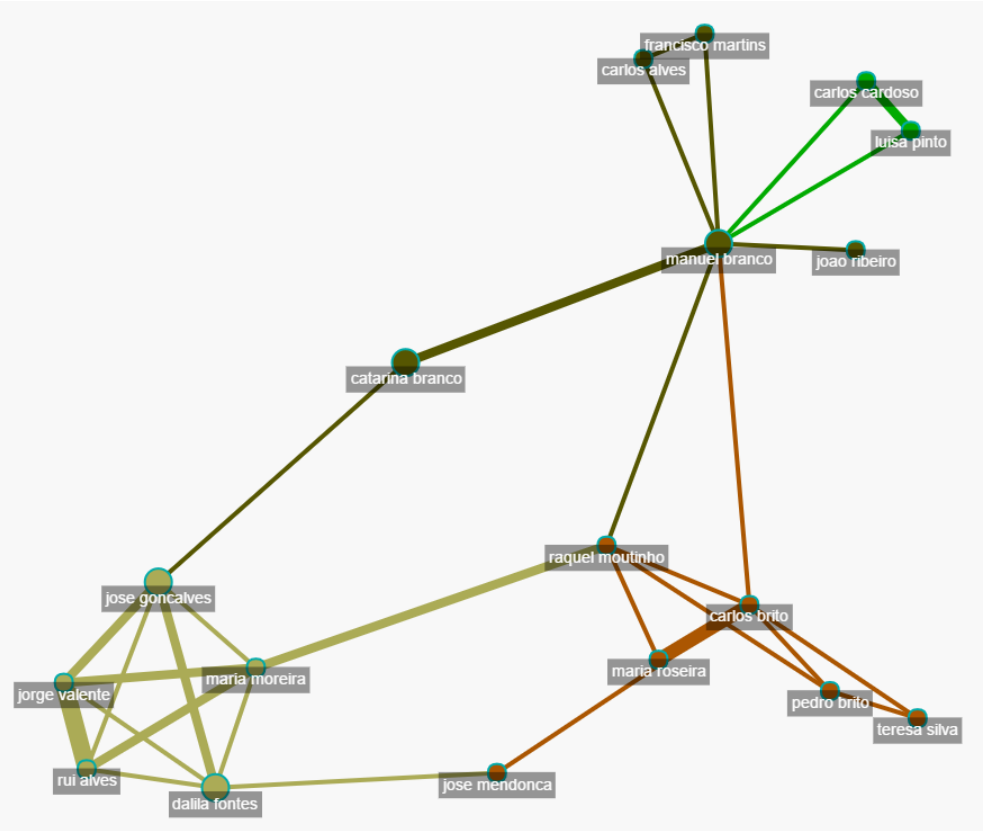

Figure B.10: Network and affinity groups for *Management*.

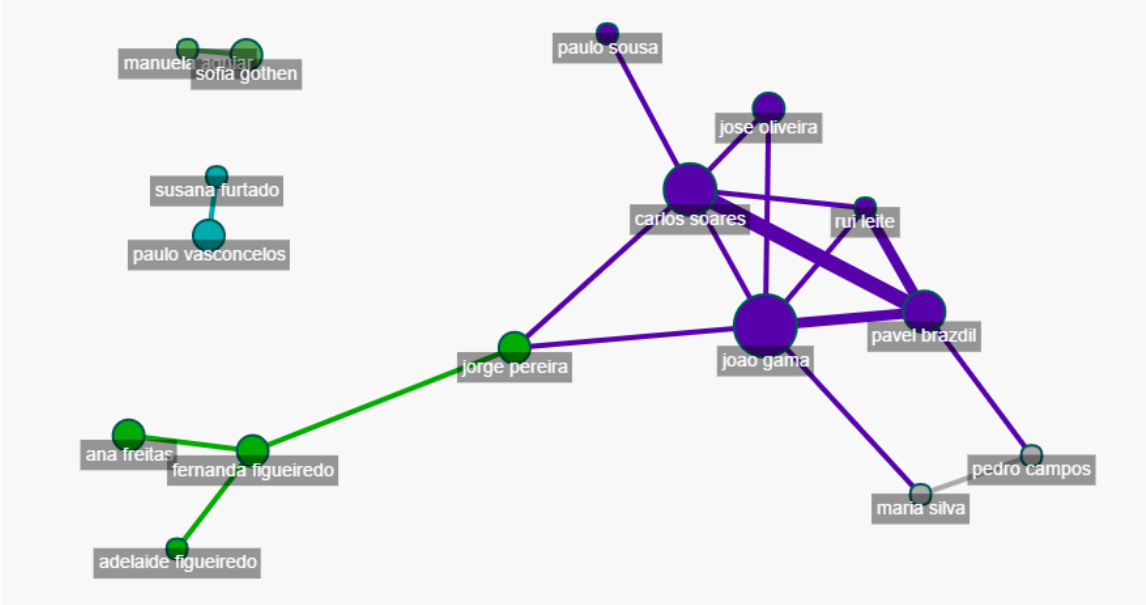

Figure B.11: Network and affinity groups for *Maths and InfSci*.