Universiteit Leiden

Opleiding Informatica

A comparison of objective functions and algorithms for network community detection

Name:	Kristi Qato
Date:	30/08/2018
1st Supervisor:	Frank Takes
2nd Supervisor:	Vincent Traag

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

A comparison of objective functions and algorithms for network community detection

Kristi Qato

September 3, 2018

Dr. Frank Takes Dr. Vincent Traag

Leiden University

Contents

Abstract

Understanding network properties is of big interest nowadays. Recently, networks have found their way into many real-world applications. One of the most striking properties is the community structure of networks, which is useful to understand the functionality and organization of these networks. We here compare two heuristic algorithms to discover communities in a network, LOUVAIN and LEIDEN. To make an accurate comparison we have included networks with ground truth communities. Therefore we have generated two synthetic networks (benchmark networks) and use three real-world networks of different sizes. We make a comparative analysis of the performance of the algorithms based on three metrics: the quality value of the objective functions, the number of communities detected by each algorithm, and the similarity measure NMI that calculates how similar the communities detected from our experiments are to the communities provided by the ground truth data. We identify which combination of algorithms and objective functions would reveal the best community structure. We have found that the best method for community detection was the Leiden algorithm, which found the best community structure according to the various quality functions.

1 Introduction

The study of networks sometimes known as graph theory has its genesis in 1736 with a famous paper [6], known to be the first use of graphs in a real-world problem. Euler used networks to show how things are connected. Over the last years, graph theory was used in innovative multidisciplinary tools, which facilitated a way to understand many real-life networks such as biological, technological and social networks [23]. In that way, scientists are using networks to represent interaction on proteins, individuals, or computers as a pair of interconnected items. Recognizing the unexpected patterns across those networks is a challenging task that has attracted a substantial amount of attention from the scientific community [12].

Networks consist of points which are called nodes or vertices, connected with each other via links or edges. These networks have some properties that depend on their topological architecture. A key property is community structure, which deals with finding a group of nodes that are densely connected with each other within the group and more sparsely with the nodes outside the group. Community detection plays an important role due to the fact that it provides useful insights into relationships and the organizational structure of networks.

Detecting the community structure of networks resulted in the development of many tools and techniques from various disciplines, such as statistical physics, biology, applied mathematics, computer science, and sociology. Their goal is to detect relevant communities while minimizing the computational complexity of the underlying algorithm [29]. The aim of our research is to a make a comparative analysis of methods for community detection in networks, in order to identify the best method that would reveal the best community structure.

Throughout this research, we will consider two heuristic algorithms which are used to reveal the community structure of networks. We will consider the traditional LOUVAIN algorithm by Blondel [3] and the LEIDEN algorithm proposed by Traag, Waltman and Eck [27]. The heuristic approach of algorithms focuses on the optimization of an objective function in order to find the community structure of the networks. An objective function is defined as a quantitative criterion to quantify the quality of the discovered community structure. For both algorithms, we will consider four objective functions (also known as quality functions), MAP EQUATION proposed by Rosvall [21], Reichardt and Bornholdt Configuration (RBC) [20], the asymptotic formulation of SURPRISE [2, 25], and the degree corrected Stochastic Block Model (SBM) [11].

Measuring the performance of our algorithms requires analyzing artificial networks or real-world networks with a well-defined community structure (known as a ground truth). For our research we are using synthetic networks (benchmark networks) proposed by Lancineti, Fortunare and Radichii (LFR) [15] and three real worlds networks: EMAIL, which represents the relationships between two persons who have exchanged at least one email and the ground truth communities representing the community structure of the network where each person is part of a department (a department represent a community). AMAZON, which represents the frequencies of bought products, and YOUTUBE which represents friendships groups which users can join. We mention that for AMAZON and YOUTUBE we are not including the ground truth communities.

Attention must also be paid to the metrics used for the comparison of the algorithms. The metrics used would differentiate the results among algorithms and functions. We are considering three metrics, firstly the quality value of the network division into communities by each objective function, indicating which of the algorithms would perform the best by scoring high values of quality measure. Secondly, the number of communities found, which we are expecting to match the number provided from the ground truth communities. Thirdly, NMI, a similarity measure. NMI is a metric widely used in community detection papers, which compares the communities found by algorithms with the ground truth communities. A high NMI indicates that the community structure detected by our algorithms match the ground truth communities. For our research we have made a comparative analysis of two algorithms namely, Louvain and Leiden, with the four objective functions map equation, rbc, surprise, sbm. This leads to the main research question:

Which combination of algorithms and objective functions gives the best division of a network into communities?

Section 2 illustrates related work, Section 3 covers the notation we are going to use in the paper and Section 4 describes the methodology which is divided into two parts: introduction of four objective functions and algorithms to. Section 5 describes the experimental setup, Section 6 summarizes the results from our experiments, Section 7 describes the conclusions.

2 Related Work

Some preliminary work in this field has been focused primarily on the graph theory introduced by Euler [6], who published the first paper in this field, entitled ”Seven Bridges of Köningsberg”. The paper provides a description of the mathematical aspects of networks and the scientist used it as an offshoot to reveal new theories about networks. Erdös and Rényi [5], used the theory of Euler to develop the probabilistic theory of networks. Eventually, the effect of the development of the probabilistic network theory inspired other scientists such as Duncan [28] and Barabasi [1], who focused on the mathematical description of different network topologies.

However, previous studies on the mathematical aspects of graph (network) theory found their application in a broad range of life aspects, such as biological, information and social networks. Investigating different network properties caught the interest of many researches like Newman [16], which made a review about different types of networks properties. Some of the most significant network properties are the small-world effect, clustering coefficient, centrality measure and community structure.

In addition, Fortunato's paper [9], describes the community structure of the networks as a modern discipline in the network theory. Fortunato provides a user guide on how to detect the communities in the networks. The demand to discover the community structure of networks was followed by the development of new methods including a variety of community detection algorithms. A lot of algorithms were introduced among scientists such are: Girvan and Newman algorithm [10], the ”Fast greedy modularity optimization” by Clauset [4], ”Fast modularity optimization”by Blondel [3] ”, the algorithm by Radicchi [19], etc.

Furthermore a lot of objective functions or quality functions were introduced. Mentioning the famous ”Modularity maximization” [17, 18], proposed by Newman. Soon, researches proved that modularity has a major drawback known as ”Resolution limit problem” [8], that prevented to find the correct structures of the networks. For further explanations the problem is illustrated on the Appendix 7. As a consequence, several objective functions were developed, such as: Expansion, Conductance, Normalized cut [22], Map equation [21], Surprise [2, 25], Reichardt and Borhnoldt Configuration [20] etc; which are thought to overcome the resolution limit of modularity. Due to the variety of algorithms and objective functions, some preliminary work was carried out to make a comparative analysis of community detection algorithm carried out from Lancichinetti and Fortunato [14].

3 Preliminaries

This section describes the notations used throughout the paper. Thus, we explain the definition of a networks, the definition of a community and the Normalized Mutual Information (NMI) similarity measure.

3.1 Network definition

A network is mathematically represented as a graph, which is a set of nodes or vertices that can be connected to each other by edges or links. A network or a graph G is a set of nodes V and a set of edges E, where $G = (V, E)$. Two nodes i and j are joined by an edge $e(i, j) \in E$. In this case the nodes i and j are said to be adjacent and edge e is incident to nodes i and j.

Additionally another notation for the definition of a network is used throughout this thesis, which is the representation of the network as an adjacency matrix. A finite graph G of n vertices can be represented by the $n \times n$ adjacency matrix $A = [A_{ij}]_{n \times n}$. An entry in A_{ij} equals 1 if there exist a link between nodes i and j, and 0 otherwise. We use n for the number of nodes and m for the number of edges.

In this research we consider undirected and unweighted networks. A network is undirected where edges or links do not have any orientation and is possible to traverse both ways, so $e(i, j) = e(j, i)$, whereas the opposite stands for the directed network.

Furthermore to define an unweighted network, first we have to define what is a weighted network. A weighted network refers to an edge-weight network, which edges have some weight or values. Those values could represent costs, lengths or capacities depending on the type of the network. Without the qualification of weighted, the networks becomes unweighted.

3.2 Community definition

A common topological property of networks is the community structure, which can be seen as a classification of objects in categories. In the paper by Radicchi [19], in a community the links within the community are denser compared to the links pointing outside of the community. Those communities are separated from the rest of the network. We mark that the terms group, module and community are used interchangeably.

A community is a subgraph $G' = (V', E')$ of the graph $G = (V, E)$ where $V' \subseteq V$ and $E' \subseteq E$, such that for all $e = (i, j) \in E'$, we have that $i, j \in V'$. When G' is a subgraph, then $G' \subseteq G$.

Quantity K_i denotes the degree of a node i, where $K_i = \sum_j \mathbf{A}_{ij}$. For a subgraph $G' \subseteq G$ and various nodes $i \in V(G')$, the degree could be split two terms: internal degree and external degree. Internal degree, K_i^{in} stands for the number of edges connecting node i to the other nodes belonging to G', and the external degree K_i^{out}, stands for the number of edges toward nodes in the rest of network. So, $K_i = K_i^{in}(G') + K_i^{out}(G')$, where $K_i^{in}(G') = \sum_{j \in V'} \mathbf{A}_{ij}$ and $K_i^{out}(G') = \sum_{j \notin V'} \mathbf{A}_{ij}$ [19].

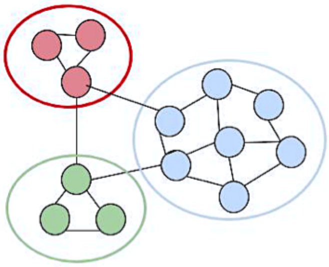


Figure 1: Communities

Two definitions of the community described are [19]:

1. Definition of community in the strong sense. A community is a group of nodes or vertices where the number of links within the community is larger than with the rest of the graph: $K_i^{in}(G') > K_i^{out}(G'), \forall i \in V'$.
2. Definition of community in the weak sense. A community is a group of nodes or vertices where the sum of all degrees within G' is larger than the sum of all degrees pointing to nodes in the rest of the network: $\sum_{i \in V'} K_i^{in}(G') > \sum_{i \in V'} K_i^{out}(G')$.

In our thesis we are working with the communities which are defined in the strong sense.

Figure 1 shows an illustration of communities. It shows a network with three groups, where the number of nodes $n = 13$ and the number of edges between pairs of nodes $m = 18$.

3.3 Normalized Mutual Information

A variety of community detection algorithms (including objective functions) has been developed, which can be used to detect the community structure of networks. Not all of them lead to the same community structure. Therefore it is necessary to compare which of the functions would find the best community structure. For this purpose, we consider the similarity measure named Normalized Mutual Information (NMI).

Normalized Mutual Information (NMI) is a similarity measure widely used to evaluate the accuracy of community detection algorithms. Unlike other similarity measures like Pearson's linear correlation coefficient that accounts for linear relationships, or rank correlation coefficients that can detect monotonic dependencies, NMI considers all types of dependencies [13]. It is a general measure which quantifies the amount of information obtained between two different networks partitioned into communities.

We denote P_R for partition R and P_S for partition S of a same network. To define NMI we calculate the marginal probability of a randomly selected node i being in group r of partition R as $P_R(r) = \frac{n_r}{n}$ and in group s of partition S as $P_S(s) = \frac{n_s}{n}$, where n_r and n_s are the number of nodes of group r and s. Then we calculate the joint probability of node i being in both groups r and s as $P_{RS}(r,s) = \frac{n_{rs}}{n}$, where n_{rs} is the number of nodes in both groups. Finally we can formulate NMI as [30]:

$$NMI(P_R, P_S) = \frac{2I(P_R, P_S)}{H(P_R) + H(P_S)}$$

where $I(P_R, P_S) = H(P_R) + H(P_{R|S})$, where $H(P_R) = -\sum_r P_R(r) \log P_R(r)$ is the entropy of distribution P_R (and analogously for P_S), and $H(P_{R|S})$ is the conditional distribution where $P_{R|S} = \frac{P_{RS}(r,s)}{P_S(s)}$. The range of values of NMI is between $[0,1]$, where 1 stands for the perfect similarity between two divisions of the network into communities and 0 stands for no similarity.

4 Methodology

This section describes the methodology of our research. We start describing four of the objective functions: MAP EQUATION, RBC, SURPRISE, SBM. Then we proceed by describing the approach of the two algorithms: LOUVAIN and LEIDEN.

4.1 Objective functions

Algorithms are supposed to identify good partitioning of the network into communities. Therefore it is necessary to have a quantitative criterion to compute the goodness of a network partitioning [7], called an objective function. All the functions described in the coming sections will be optimized by the two algorithms described in Section 4.2. Their optimization reveals the community structure of a network. Each objective function takes as parameter a network and the outputs consist of the community structure of the network.

4.1.1 Map equation

MAP EQUATION is an objective function proposed by Rosvall and Axelsson [21], that highlights different aspects of the network's structure. Differently from other objective functions that uncover the community structure based on the formation of the network, MAP EQUATION searches for structures in the network that are relevant with respect to how the information flows through it. It is a flow-based approach, which focuses on the system behavior of the network.

Considering MAP EQUATION, optimizing it would result in shortest description length of the flows on the network, picking one that gives the shortest length. In order to achieve the shortest description MAP EQUATION sets a theoretical limit of how concisely we specify a network path using given communities. To find the community structure the function looks for a community structure of n nodes in q modules (communities). For undirected weighted networks, Rosvall and Axelsson define it as [21]:

$$Q_{\text{MAP EQUATION}} = w_{\curvearrowright}\log(w_{\curvearrowright}) - 2\sum_{c=1}^{q} w_c \curvearrowright \log(w_c \curvearrowright) - \sum_{i=1}^{n} w_i \log(w_i) +$$

$$\sum_{c=1}^{m}(w_c \curvearrowright + w_c) log(w_c \curvearrowright + w_c).$$

Throughout our thesis we calculate MAP EQUATION for unweighted networks. For undirected unweighted networks, w_i of node i is the total number of links connected to that node. We calculate the total number of links as a fraction of the number of links connected to a node (or a community), divide by twice the total number of links in the network. Here, w_i is the total number of links of node i, then $w_c = \sum_{i \in c} w_i$ is the total number of links inside the community c. Then $w_c \curvearrowright$ is the total number of link exiting community c. Ending with $w_{\curvearrowright}$ which is the total number of links between the communities in the network. When we want to divide the initial network into communities we only have to track the changes in $w_c \curvearrowright$.

4.1.2 Reichardt and Bornholdt Configuration (RBC)

Reichardt and Bornholdt Configuration (RBC) is an objective function proposed by Reichardt and Bornholdt [20], based on principles of statistical mechanics. They interpret the process of community detection as finding the ground state of an infinite range spin glass. While using a spin model to capture the community structure of the network, the energies are interpreted as objective functions. Thus, the energy of the spin system is comparable to the objective function that corresponds to the number of the communities found as the number of occupied spin states.

The objective function for an unweighted, undirected network, uses a linear resolution parameter [20]:

$$Q_{\text{RBC}} = \sum_{ij} \big(A_{ij} - \gamma \frac{K_i K_j}{2m}\big)\delta(s_i, s_j)$$

where A_{ij} is the adjacency matrix, K_i is the degree of node i, m is the total number of edges, s_i denotes the community of node i, $\delta(s_i, s_j) = 1$ if $s_i = s_j$ and 0 otherwise. The number of spin states determines the maximum number of communities allowed and it could be as large as n, and γ is a resolution parameter [2]. The parameter is depended in the number of links and the network size. The resolution parameter prevents detecting smaller communities in large networks. Also it used to detect community structures at different hierarchical levels. According to the [24] we note that in our function we are setting the parameter $\gamma = 1$, which guaranties that each community consist of a number of group of nodes, not only to a node .

4.1.3 Asymptotic Surprise

SURPRISE is a quality function that measures the possible ways of sampling edges between two nodes, whereof some are internal edges. It is a method mostly focused on classical probability [2, 25]. The asymptotic formulation of the function evaluates both the number of links and units in each community. The function assumes that even though the graph grows the number of internal edges the number of expected internal edges would remain fixed. The function is defined as KL divergence measure [25]:

$$Q_{\text{SURPRISE}} = mD(q||\langle q \rangle).$$

Here we adopt a slightly different notation where $q = \frac{\sum_c m_c}{m}$is the fraction of internal edges in the community c, $\langle q \rangle = \frac{\sum_c \binom{n_c}{2}}{\binom{n}{2}}$, is the expected fraction of internal edges, where n_c is the number of the nodes in community c. SURPRISE is formulated as KL (known as Kulllback-Leibler divergence) [24]:

$$D(x||y) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}.$$

4.1.4 Degree corrected Stochastic Block Model

Stochastic Block Model (SBM) [11] is a model for generating synthetic networks, with a known block (division into groups or communities). SBM consists on fitting block models to empirical networks data as way to discover the community structure of the network. It is referred as a posteriori block modeling.

Furthermore, in our research, we are using a different formulation of the standard Stochastic Block Model as a result of which the classic SBM splits the network into groups with high and low degree, whereas, this approach includes heterogeneity in the degrees of nodes which improves the results. This is called the degree corrected stochastic block model. The degree corrected SBM with degree corrected performs better than the classic SBM. The quality function is formulated as [11]:

$$Q_{sbm} = \sum_{rs} m_{rs} log \frac{m_{rs}}{K_r K_s}$$

where r and s stand for two different communities, m_{rs} is the number of links between those two communities, K_r and K_s stand for the total degree (the sum of node degrees) of two communities. The formulation gives an unnormalized log-likelihood of the objective function.

We have to note that compared other functions SBM takes a parameter q of a fixed number of communities. This is used to divide the nodes into communities such that these assignments maximize the likelihood of the model according to the observed edges. The problem of SBM is that the quality defined in the formula explained above is always higher when more communities are assigned. So if the number of communities is not defined, it will end up with a single partition. In real-world networks where the number of communities is not known, degree corrected SBM would be inefficient. That is the purpose why we have not tested SBM for networks where ground truth communities is not known.

4.2 Algorithms

This section describes the algorithmic approaches. First we describe the Louvain algorithm and secondly the Leiden algorithm.

4.2.1 Louvain Algorithm

The first algorithm we are going to test is the Louvain algorithm introduced by Blondel [3]. It follows a local moving heuristic approach, which continuously searches for an improvement on the objective function by moving nodes from its current community to a different community.

The algorithm has a simple formulation. It has two phases, where each phase is repeated iteratively. The first phase consists of the local moving of nodes between communities. The second phase is the aggregation, which builds a new network by assigning the nodes in the same community as a single node. After the second phase is completed the algorithm repeats itself, but now instead of the single node communities it starts from the community structure built in the second phase.

In the first phase, we assign each node to a community, which implies that we have as many communities as there are nodes. Subsequently, the algorithm uses the local moving heuristic, where for each node i, it finds its neighbors j. The algorithm evaluates the highest improvement on the quality value of each of the given objective functions by removing i from its community and by placing it to the community of j. A node only moves if the quality value strictly improves; if not the node i stays in its community [3]. The first phase continues until there are no further improvements on the quality value. We note that the difference on the quality value on the move node phase for each objective function is calculated on the Appendix 7.

After the first phase has been successfully completed, the algorithm continues with the second phase, the aggregation of the network. This results in a new network where nodes are the communities found in the first phase. After two phases are completed, we call it a pass. Afterwards the algorithm starts to repeat itself, where the local moving heuristic is applied again but this time in the reduced network. The algorithm stops when a network is obtained that cannot be aggregated further.

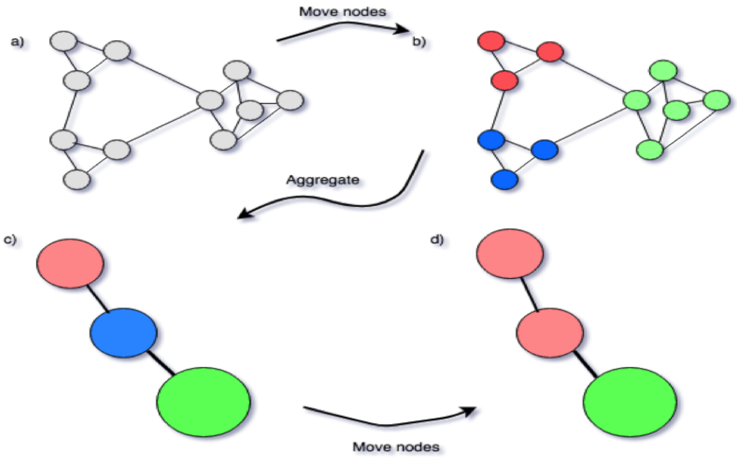


Figure 2: LOUVAIN algorithm

Figure 2 illustrates all the phases of the algorithm. It starts with a singleton community where each node is a community (a). The algorithm moves around nodes to find a community (b) and aggregates it at (c). Once all the phases of the algorithms are completed, it repeats the two phases on the aggregated graph [27].

However, it was shown that the LOUVAIN algorithm has some limitations. The algorithm may generate badly connected communities. As stated [27], LOUVAIN may return communities that are internally disconnected, meaning that a node may be moved to another community while it may have acted as a bridge between different nodes of its old community. Removing that node from its old community will disconnect the old community. Running the LOUVAIN algorithm actually worsens this problem, although it does improve the quality of the network partition into communities.

4.2.2 Leiden Algorithm

The second algorithm is the LEIDEN algorithm which overcomes the main problem of LOUVAIN. In contrast to the LOUVAIN algorithm, LEIDEN guarantees that the communities are connected.

We consider LEIDEN as a solution to the aforementioned LOUVAIN problem. The algorithm consists of three phases: firstly the local moving heuristic of nodes, secondly, the refinements of the communities and thirdly, aggregation of the network. Different from classical LOUVAIN, LEIDEN has the refinement phase, whose purpose is to identify possible sub-communities before aggregating the network. Thus, instead of moving around communities after aggregation, we can move around sub-communities after aggregation.

The refinement phase [27] starts with a singleton partition, where each node is in its own community. Within each community of the non-refined partition, we locally merge nodes in the refined partition: nodes that are in their own community in the non-refined partition can be moved to another community, therefore a community in the non-refined partition could be split into multiple communities in the refined partition. We only merge nodes in the refined phase when both parts are well connected to their community in the non-refined communities. Then the refined partition is used for aggregation of the network. We must also point out that in the refinement phase nodes are not necessarily merged with the community that has the largest improvement in the quality function. Alternatively, a node could be merged with any neighboring community that improves the quality function.

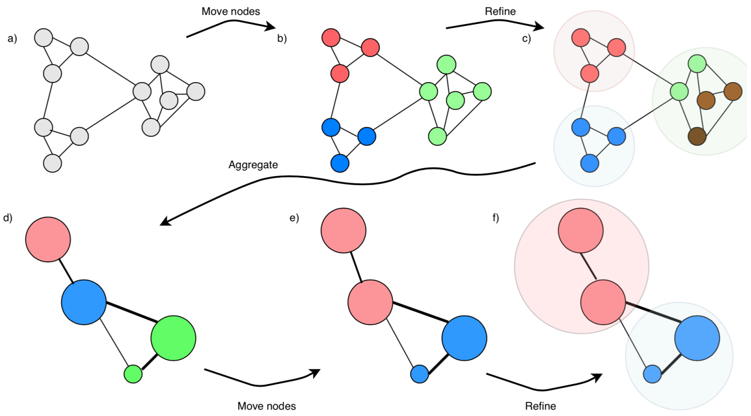


Figure 3: Leiden algorithm

Figure 3 explains the phases of Leiden algorithm. It starts from a singleton partition (a), the algorithm moves around nodes to find a partition (b), which is then refined (c). The aggregated network (d) is based on the refined partitions(communities). For example, the green community in (b) is refined in two sub communities in (c), which after aggregation becomes two separate nodes in (d), while both are part of the green community. The algorithm then finds partitioning (e), which remains the same after refinement (f). This is repeated until the quality function cannot be improved further.

Also, Leiden has been shown to be faster than Louvain in terms of computational time. The difference is more significant in large networks, being up to 20 times faster in real-world networks [27].

5 Experimental setup

In this section, we introduce the metrics we are using in our experiments and the networks that will be tested by our algorithms.

5.1 Description of experiments

This part explains all the metrics we are using to compare the performance of our methods: the algorithms (Leiden and Louvain) and the objective functions (map equation, rbc, surprise and sbm). The hardware used is a MacBook-Pro with a 2,6 GHz Intel Core i5 processor and a memory of 8 GB. Also, the experiments evaluation would include networks with ground truth communities to make more accurate comparison of the three metrics being used throughout this research. Especially they will be used to compute the similarity measure NMI, which is an important metric for the comparison between two different networks partitioned into communities.

Firstly we are going to test our methods based on the quality measure. For each network we are going to measure the ratio of the quality value uncovered from each algorithm and objective function and the quality of the ground truth communities provided for each network. We are calculating the ratio in order to make the difference in the quality more comparable. The ratio should be close to 1 if the community detection works well, or greater than 1 if the quality uncovered from our algorithms is higher than the quality of the ground truth communities.

Secondly we would measure the number of communities detected from our algorithms and objective functions. The number of communities found should match the number of communities provided in the ground truth communities (GT), allowing us to estimate how accurately the algorithms performed.

Thirdly we calculate the similarity measure NMI, which is explained in Section 3.3. NMI is applied to asses to what extent the communities found from match the ground truth communities. Thus, we would have a more accurate picture of how well our methods performed. We emphasize that for ground truths which have overlapping communities, where one node could be part of one or more communities, NMI is not calculated. NMI is not well-defined for overlapping communities, and our algorithms do not find overlapping communities.

Furthermore, as a limitation to our experiments, we note that for SBM, the calculation of the quality measure, the number of communities and NMI, are not possible for large networks, because of the large computational time.

We replicated our experiments 5 times and took the average of the quality score and the NMI.

5.2 Benchmark networks

To test the algorithms and the objective functions, we require networks which have well-defined community structure. We are using synthetic networks (benchmark networks) for our experiments, due to the fact that there are not many accessible real-world networks with predefined community structure.

5.2.1 Benchmark parameters

We will use the benchmark suggested by Lancichinetti, Fortunato, and Radicchi (LFR) [15], generating undirected, unweighted networks. LFR is used extensively among scientists to test algorithms for community detection, as it has many realistic properties, similar to real-world networks. The benchmark accounts for heterogeneity in the node degrees and community size. Heterogeneity stands for the variety of the node degree in the structure of the network, confirming that many real-world networks have a skewed distribution. Also it generates communities with different sizes, which is very common in real world networks. LFR benchmark networks are suitable for testing community detection algorithms as they can be constructed very quickly, and can span several orders of magnitude in network size.

The construction of the benchmark networks includes parameters [15]:

1. Each node has a degree taken from a power law distribution with exponent γ. The extremes of the distribution, the minimum degree, K_{min} and maximum degree, K_{max}, are selected such that the average degree is $\langle K \rangle$.

2. We define a mixing parameter μ which assigns the links for each node. Each node shares a fraction $1 - \mu$ of its links with the other nodes in its community and a fraction μ with the other nodes in the network.

3. The size of communities are taken from a power law distribution with exponent β such that the sum of all communities matches the number of nodes in the network.

The benchmark is generated so that in the beginning all the nodes are considered to be single community. In the first iteration every node is part of a randomly chosen community. Then if the community size is greater than the internal degree of the nodes then the nodes become part of the community, if not then the node remains in its own community. The procedure stops when there are no more single nodes left [15].

5.2.2 Benchmark parameter settings

The parameter settings for our benchmark networks are:

1. We are generating benchmark networks of different sizes. We start with a benchmark network of $1000, 5000, 100, 000$ and $500, 000$ nodes. The purpose of generating different sizes of benchmark networks is that we want

to understand how our algorithms and objective functions perform in large and small networks.

2. Degree distributions often follow a power law in real networks. The realistic range of the power law exponent falls between 2 and 3. For our experiments, we have set the degree distribution $\gamma = 2$, because for a value in the range between 2 and 3, the parameter values has a negligible effect for the task of community detection. The maximum degree is related to the networks size, so we have set the parameters of $K_{max} = 50$ and $\langle K \rangle = 15$ fixed for all the benchmark networks.

3. The mixing parameter is known to be the most influential parameter in the generation of the benchmark. It is used to generate the community structure of the ground truth communities. For our experiment we have used different values of the mixing parameter, ranging from $\mu = 0.2$ to $\mu = 0.9$. A low mixing parameter indicates a clear community structure because there are only few links between communities, which makes the communities clearly separated. As μ increases the proportion of intercommunity links becomes higher making community identification a more difficult task.

4. The community size distribution is set to $\beta = 1$, which is similar to real-world networks. As in the case of the degree distribution, the changes in β between the values 1 and 2 have a negligible effect on the results for community detection.

5.3 Real-world networks

For our research we have also included real-world networks with ground truth communities. Even though artificial networks seems to be an appropriate alternative to test community detection algorithms, we can never be completely assured that the generated networks are perfectly realistic. The networks we are going to use are from the Stanford data collection (`https:`

//snap.stanford.edu/data/). In table 1 we summarize the properties of the networks such are the number of nodes, number of edged and the average clustering coefficient. Clustering coefficient is a measure of the degree to with the nodes in the network tend to cluster together.

- The first real-world network is EMAIL, which was generated using email data from a large European research institution. It is an unweighted and undirected network. An edge exists between two persons if they have ever exchanged an email with each other. The network also includes ground-truth communities. A community represents one of the 42 departments at the research institute. It contains 1,005 nodes and 25,571 edges.
- The second network we are going to use is AMAZON: it has 334,863 nodes and 925,872 edges. It represents the AMAZON product network. The network describes the frequency of bought products. If a product i is frequently co-purchased with product j, an edge exists between two products. Even though the network is weighted, for our experiments we are using it as an unweighted network, because of MAP EQUATION which is not applicable to weighted networks. For the AMAZON network, we are not considering the ground truth as it has overlapping communities.
- The third network is YOUTUBE. The network describes the users as nodes, and users can form friendships (edges) with the other users. It has 1,113,890 nodes and 2,987,624. For the YOUTUBE network, we are also not considering the ground truth communities, because of overlapping communities.

	Nodes	Edges	Average clustering coefficient
Email	1005	$25,571$	0.3994
Amazon	$334,863$	$925,872$	0.3967
Youtube	$1,113,890$	$2,987,624$	0.0808

Table 1: Network properties

6 Results

In this section we provide the results on the performance of the algorithms and objective functions. First we describe the results from the benchmarks in terms of the ratio of the quality measure, the number of communities found and the NMI. Secondly we describe the results on real-world networks.

6.1 Benchmark networks results

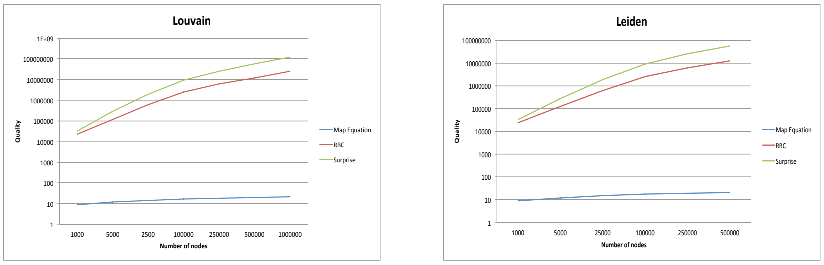
This section demonstrates the performance of the experiments on the benchmark, beginning with the quality measure,the number of communities, finishing with NMI.

6.1.1 Results on quality measure and number of communities

The experiments have been carried out on the LFR benchmark described in Section 5.2. We have included different sizes of the benchmark networks. We would like to emphasize that we are using a range of values of mixing parameter $\mu = [0.2; 0.9]$, which consists of different community structure for the ground truth data. Even though all the results depend on all parameters of the benchmark, we are setting the parameter $\gamma = 2$ and $\beta = 1$ as constants.

Moreover, we tested the benchmark networks on two algorithms the Louvian and Leiden algorithm, including the four objective functions map equa-

TION, RBC, SURPRISE and SBM. We ran the algorithms five times and took the average of the quality value of each objective function. For the quality measure we measured the ratio of the quality value detected from our algorithms and the quality value of the ground truth communities. The ratio would imply how similar the score on the quality value is compared with the ground truth. Yet, for the comparison of the number of the communities found we have included the number of the ground truth communities as GT in our charts.



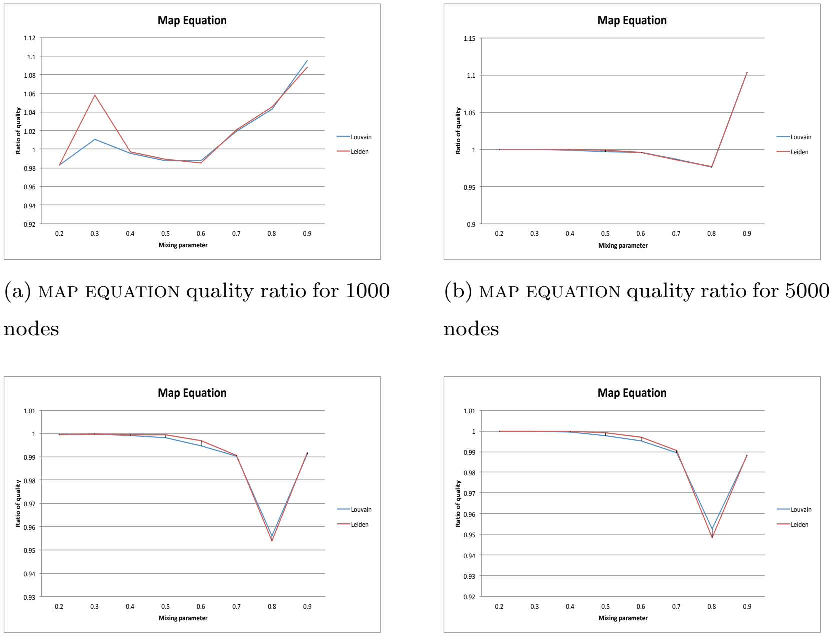
(a) LOUVAIN quality measure

(b) LEIDEN quality measure

Figure 4: Quality measure

In Figure 4, we illustrate on a logarithmic scale the quality measure as the number of nodes in the benchmark increases. We remark that not all functions have the same scale on the quality value (especially MAP EQUATION) but we want to emphasize that all objective functions scales with the number of nodes/edges. SBM is not included as we were not able to measure it in large networks. SBM is available only for the benchmark of 1000 and 5000 nodes.

We begin our comparative analysis with the benchmark of 1000 nodes until $500,000$ nodes.



(a) MAP EQUATION quality ratio for 1000 nodes

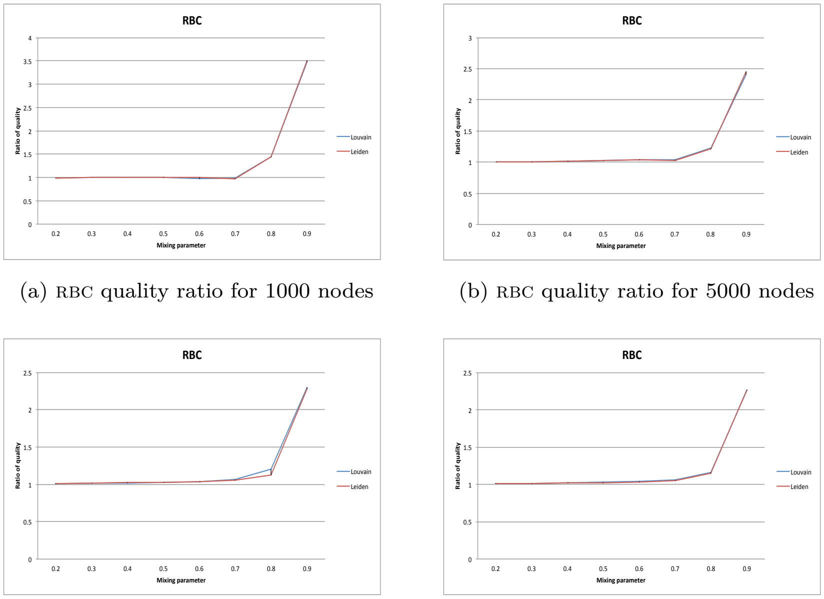
(b) MAP EQUATION quality ratio for 5000 nodes

(c) MAP EQUATION quality ratio for 100,000 nodes

(d) MAP EQUATION quality ratio for 500,000 nodes

Figure 5: Quality measure

In Figure 5 we illustrate the result of the quality measure of MAP EQUATION. In all benchmark networks LEIDEN algorithm seems to have better results than LOUVAIN. The ratio of the quality is close to 1 for the mixing parameter $\mu = 0.2$ and this is significant for larger benchmark networks of $100,000$ and $500,000$ nodes.



(a) RBC quality ratio for 1000 nodes

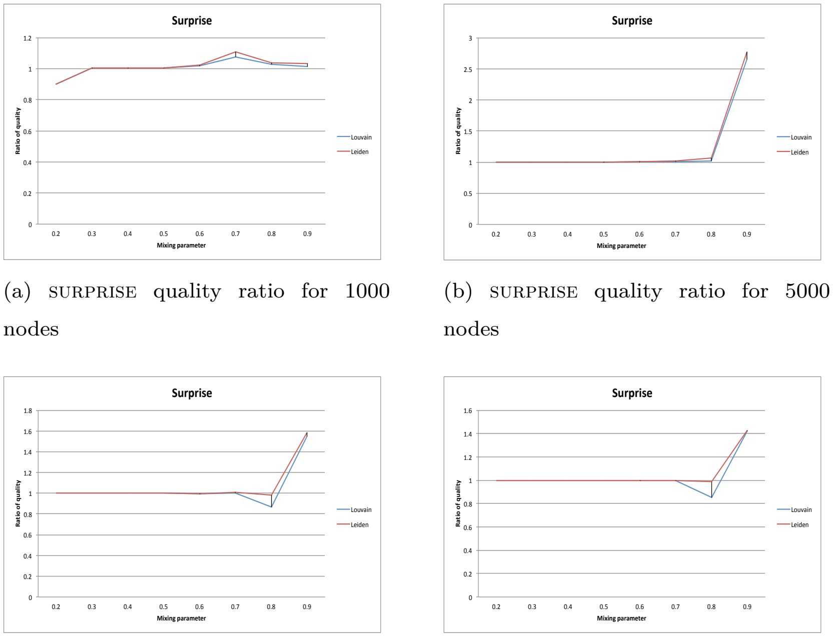
(b) RBC quality ratio for 5000 nodes

(c) RBC quality ratio for 100,000 nodes

(d) RBC quality ratio for 500,000 nodes

Figure 6: Quality measure

However, from Figure 6 we can not say the same for RBC, where for some values of μ LOUVAIN has better results than LEIDEN. The difference in both algorithms is so small that is unobservable from the charts. Moreover, we infer that the ratio of the quality measure is mostly 1 for all the benchmark networks, except case of MAP EQUATION. If the ratio equals 1, then this implies that the quality value uncovered from our algorithms is equal with the quality value of the ground truth communities.



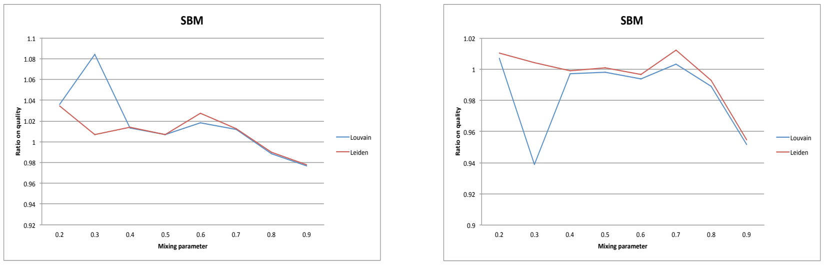
(a) SURPRISE quality ratio for 1000 nodes

(b) SURPRISE quality ratio for 5000 nodes

(c) SURPRISE quality ratio for 100,000 nodes

(d) SURPRISE quality ratio for 500,000 nodes

Figure 7: Quality measure



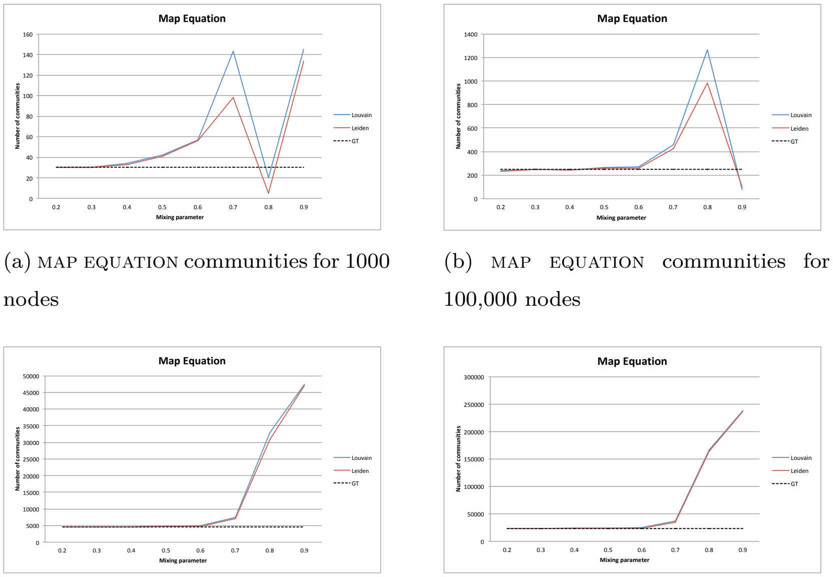
(a) SBM quality ratio for 1000 nodes

(b) SBM quality ratio for 5000 nodes

Figure 8: Quality measure

Figures 7 and 8 outline the results of the quality measure for SURPRISE and SBM. Similar to MAP EQUATION, SURPRISE scored higher values on the quality measure with the LEIDEN algorithm compared to the LOUVAIN algorithm. In common with RBC the ratio is equal to 1 for most of the values of μ, which means that the functions RBC and SURPRISE find the community structure of the benchmark networks, similar to the ground truth communities.

Differently from other functions, we want to point out the performance of SBM, which has some limitation. As we have explained in Section 4.4, SBM takes a parameter of the predefined number of communities, which in our case would always be 30 for the benchmarks of 1000 nodes and 250 for the benchmark of 5000 nodes. That parameter also influences the quality measure of SBM, because every time we generate SBM it consists of a different partition of the network. Still, that does not mean that SBM does not reveal the community structure of the network. The score on the quality ratio for SBM seems to be unpredictable. In the benchmark of 1000 nodes it has scored higher in the LOUVAIN algorithm, meanwhile for the benchmark network of 5000 nodes is the opposite.



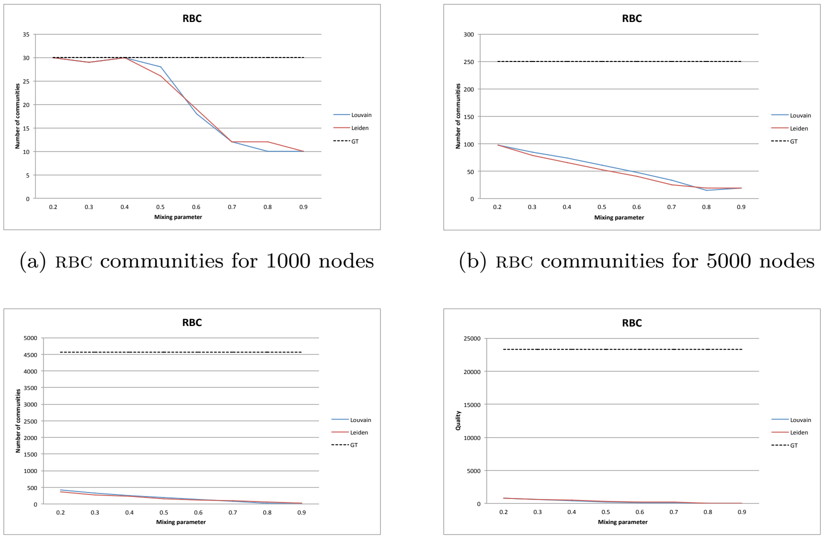
(a) MAP EQUATION communities for 1000 nodes

(b) MAP EQUATION communities for 100,000 nodes

(c) MAP EQUATION communities for 1000 nodes

(d) MAP EQUATION communities for 100,000 nodes

Figure 9: Number of communities



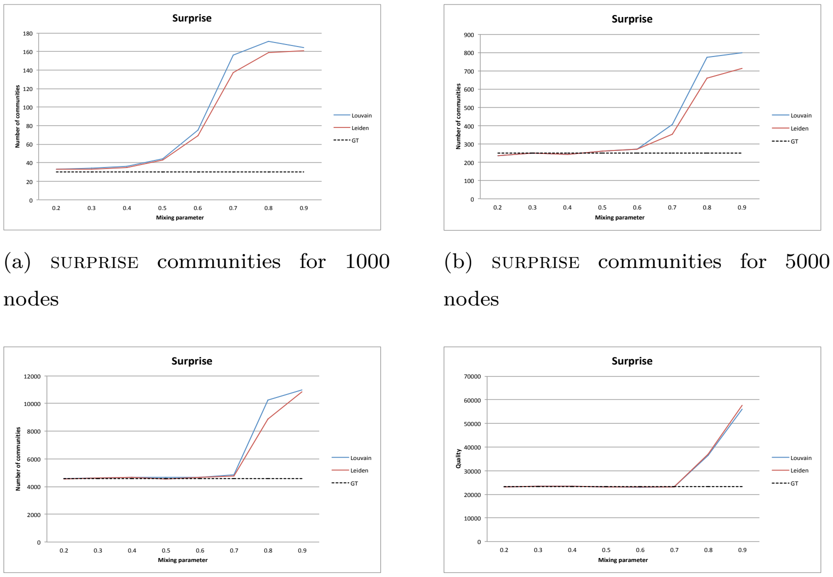
(a) RBC communities for 1000 nodes

(b) RBC communities for 5000 nodes

(c) RBC communities for 1000 nodes

(d) RBC communities for 100,000 nodes

Figure 10: Number of communities



(a) SURPRISE communities for 1000 nodes

(b) SURPRISE communities for 5000 nodes

(c) SURPRISE communities for 100,000 nodes

(d) SURPRISE communities for 500,000 nodes

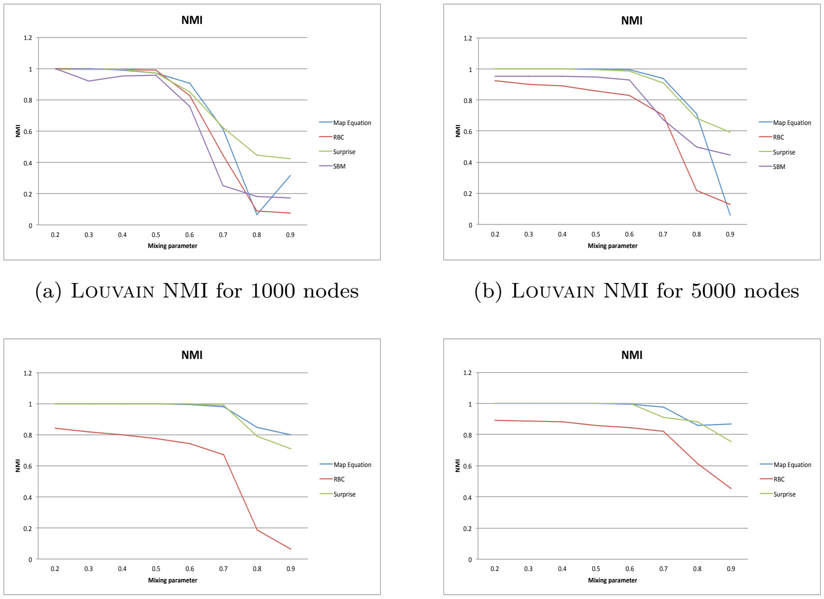
Figure 11: Number of communities

Figures 9, 10 and 11 measure the number of communities found in the benchmark networks. We see that for the small benchmark networks, where the mixing parameter $\mu = 0.2$, the methods found the correct number of communities compared to the ground truth communities. MAP EQUATION and SURPRISE detected the exact number of communities, yet we can not say the same for RBC. For benchmark networks larger than 1000 the function is not accurate.

In Figure 10 we can easily distinguish the difference between the ground truth (GT) and the communities uncovered by our algorithms. That comes as the problem of the resolution parameter, which prevents the algorithm to find communities smaller than a scale, and for the large benchmark networks this problem seems to be more significant.

6.1.2 Results on NMI

In the previous section we described the ratio of the quality measure of communities discovered by both algorithms. In this part, we will investigate the similarity of the communities found compared to the ground truth communities. As described in Section 3.3 we are measuring the similarity based on NMI.



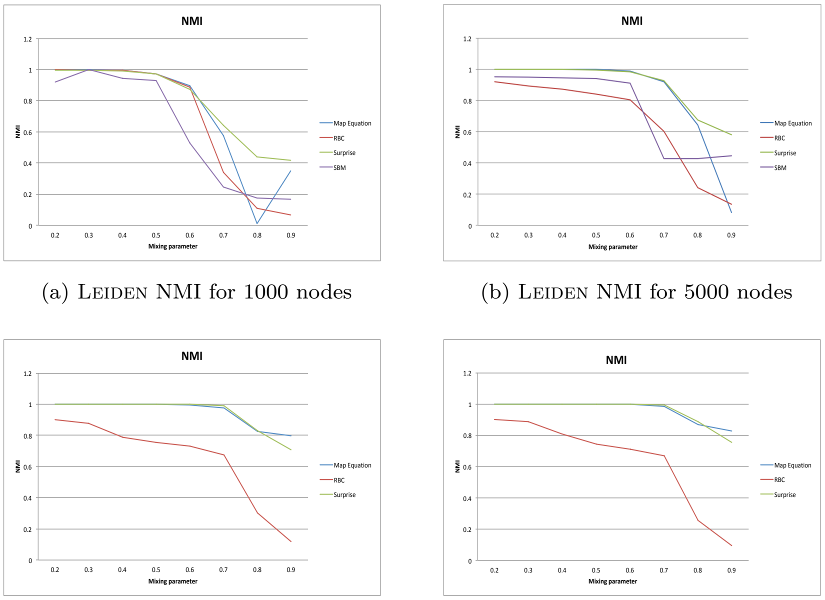
(a) Louvain NMI for 1000 nodes

(b) Louvain NMI for 5000 nodes

(c) Louvain NMI for 100,000 nodes

(d) Louvain NMI for 500,000 nodes

Figure 12: NMI measure



(a) Leiden NMI for 1000 nodes

(b) Leiden NMI for 5000 nodes

(c) Leiden NMI for 100,000 nodes

(d) Leiden NMI for 500,000 nodes

Figure 13: NMI measure

Figure 12, shows the NMI values for the LOUVAIN algorithm for all objective functions for each of the benchmark networks. We show that in general all the functions scored high on NMI except the RBC. RBC for large benchmark networks has the lowest value as a consequence of the resolution parameter. It is clearly visible that SURPRISE is the most stable function because even for large μ the NMI is almost 1.

Figure 13 shows the NMI values for the LEIDEN algorithm. Like LOUVAIN the functions have scored high values of the NMI. Similar to the case of LOUVAIN, RBC has scored low values of the NMI. We infer that all the algorithms and functions, except RBC, found comparable communities with the ground truth, up to $\mu = 0.5$.

6.2 Real-world networks results

This section describes the results form the real-world networks including EMAIL, AMAZON, and YOUTUBE. The experiments follow the same order as in the section on the benchmark networks results.

6.2.1 Results on quality measure and number of communities

We display the performance of the experiments on real-world networks as they tend to have a more complex structure in comparison with the benchmarks. We are going to show the experiments on quality measure and the number of the communities for the three real-world networks. We begin with the smallest network EMAIL, continuing with the largest that is YOUTUBE.

	LOUVAIN	LEIDEN
MAP EQUATION	6.711	$6,718$
RBC	22512.024	22511.046
SURPRISE	$5449,916$	$5449,916$
SBM	-503423.179	$-502853,777$

Table 2: EMAIL quality measure

	LOUVAIN	LEIDEN
MAP EQUATION	92	84
RBC	27	27
SURPRISE	1004	1004
SBM	42	42

Table 3: EMAIL number of communities

Table 2 shows the results of the EMAIL network. Clearly MAP EQUATION and SBM performed better on the LEIDEN algorithm, whereases we could not say the same for RBC, which scored low values of quality for the LEIDEN and SURPRISE, which shows no difference between both algorithms.

Furthermore, Table 3 indicates the number of communities found by each algorithm. We remark that SBM has the same number of communities because they are predefined. SURPRISE surprisingly found the same number of communities in both algorithms, which explains the value on the quality measure. Yet, MAP EQUATION seems to be the one that has the best results so far on real-world networks.

	Louvain	Leiden
map equation	18.793	18.865
rbc	1721238	1728799
surprise	6552873	6577454

Table 4: Amazon quality measure

	Louvain	Leiden
map equation	46812	44779
rbc	252	421
surprise	18057	18087

Table 5: Amazon number of communities

Table 4 and 5 summarize the results of the Amazon network. Obviously, all objective functions seem to reach their optimum on the Leiden algorithm which was not the same in the Email network. The values of the quality of rbc and surprise are higher in Leiden than in Louvain.

	Louvain	Leiden
map equation	15.340	15.359e
rbc	4322403	4371268
surprise	11990201	11861928

Table 6: Youtube quality measure

	Louvain	Leiden
map equation	218155	215833
rbc	7734	3675
surprise	179857	152173

Table 7: Youtube number of communities

Tables 6 and 7 show the results on the Youtube network. The results match those for Amazon, except rbc which has scored low values of quality with the Leiden algorithm.

6.2.2 Results on NMI

In this section, we show the results on the similarity measure NMI for the Email network. As explained in Section 5.3, we are not testing Amazon and Youtube as they have overlapping communities.

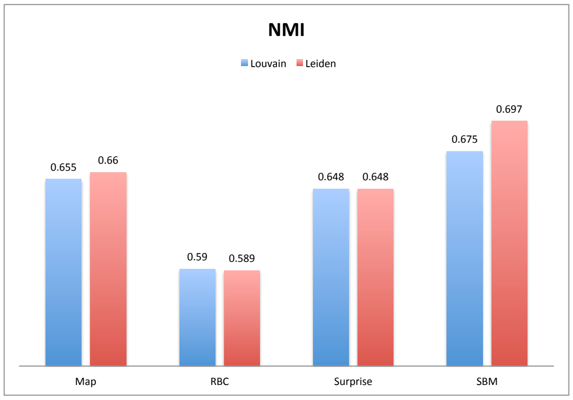


Figure 14: NMI of Email

Figure 6.2.2 shows the NMI score of the functions on the Email network. In general, all of them have scored an NMI above 0.5. The results highlight that sbm scored the highest on NMI, because we predefine the number of communities we want as $q = 42$, which equals the number of communities in the ground truth.

However, from the comparison of other functions which reveal the communities while optimizing them, we conclude that MAP EQUATION has the best similarity value. MAP EQUATION out-performed all the other functions, especially with the LEIDEN algorithm. RBC is the worst performer according to the NMI metric.

7 Conclusion

We made a comparative analysis of the performance of two algorithms (LEIDEN and LOUVAIN) for community detection and four objective functions (MAP EQUATION, RBC, SUPRIRSE and SBM). We focused on the ratio of the quality measure between the value scored by our algorithms and the value of the ground truth communities, the number of the communities found and the similarity measure NMI. The experiments have been conducted on LFR benchmark networks and in three real-world networks (EMAIL, AMAZON and YOUTUBE). The networks have been interpreted as undirected and unweighted.

As a result from the experiments on the quality measure, we conclude that the LEIDEN algorithm outperforms LOUVAIN. We found that in most of the benchmark networks and real-world networks the ratio of the quality of the LEIDEN algorithm scored close to 1, which emphasizes that the value detected from our algorithms matches the values of the ground truth communities. Even though the difference compared to the LOUVAIN algorithm was not that big, we conclude that the community structure of the networks detected from the LEIDEN algorithm is as good as the ground truth communities.

In our research we found that the most stable objective functions were SURPRISE and MAP EQUATION as they performed very well in benchmark networks and in real-world networks. Benchmark networks and real-world networks scored almost equal with respect to the number of communities in the ground truth data. The only objective function that did not detect the correct number of communities was RBC because of the resolution parameter. Yet, we want to add that the best results of each objective function were detected while they were part of the LEIDEN algorithm.

Another result from NMI is that MAP EQUATION and SURPRISE while being part of LEIDEN algorithm are the best performer among all other objective functions. The values on the NMI are the highest meaning that they found communities that are similar to the ground truth communities.

Moreover, SBM is inefficient to optimize for networks where the ground truth is not known. That as a result of the q parameter, the number of communities we want to reveal, which for large networks takes a lot of computation time to determine.

The experiments conducted in small benchmark networks of 1000 and 5000 nodes and in the EMAIL network, resulted in comparable values in all metrics in both algorithms, LOUVAIN and LEIDEN. The values on the NMI and the quality value, for the EMAIL network showed that SBM had found a better community structure with the LEIDEN algorithm than LOUVAIN. Therefore we say that SBM performed better as part of LEIDEN algorithm.

Some limitation should be noted. The limitation we had due to unavailability of large networks with ground truth data that do not have overlapping communities, made the calculation of NMI impossible. Hence, we are not sure how comparable the communities found from our algorithms are with the ground truth communities. Also, we were not capable to calculate the SBM for large networks, as a consequence of the parameter q and the large computational time it takes to detect the community structure of the networks.

Future work is required to answer our limitations. Firstly we suggest to test SBM in larger networks. Secondly, testing all the methods including algorithms and objective functions in large real-world networks with ground truth communities.

Appendix A

Here we explain the difference on the quality when we want to move nodes from an old community to a new community. This is important to highlight, because the algorithms disused in the Section 4.2 constantly look for the improvement of the quality. Therefore, optimizing the community structure of the given networks goes along with the optimization of the quality function or objective function itself.

We denote the old community with r and the new community with s. The difference in change would consist with the delta on the quality measure for each function described in Section 4.1. For each of the objective functions the difference is:

- **Map Equation** [21]:

  $$\triangle Q_{\text{MAP EQUATION}} = [w'_{\curvearrowright} log(w'_{\curvearrowright)} - w_{\curvearrowright} log(w_{\curvearrowright})] - 2\sum_{c=1}^{m}[w'_c \curvearrowright log(w'_c \curvearrowright) - w_c \curvearrowright log(w_c \curvearrowright)] + \sum_{c=1}^{m}[(w'_c \curvearrowright +w'_c) log(w'_c \curvearrowright +w'_c) - (w_c \curvearrowright +w_c) log(w_c \curvearrowright +w_c$$

  The first terms that includes the sing $(')$ calculates the difference of the links between the communities after moving node i to community s and before when node i was part of community r, followed by the difference of the exit links, and the last terms equals the difference of the exit links of the community plus the strength of node i.

- **RBC** [20]: $\Delta Q_{\text{RBC}} = K_{ir} - K_{is} - \frac{\gamma}{2m} K_i(K_r - K_i - K_s)$

  Where K_r and K_s is the sum of the degrees of nodes in the old community and new community, K_i the degree of the node i, K_{is} the number of links between i and new community and K_{ir} the number of links between the i and the old community.

- **Surprise** [2, 25]:

  $$\Delta Q_{\textsc{surprise}} = \left[D'(q'||\langle q'\rangle) - D(q||\langle q\rangle)\right]$$

  The sing ($'$) stand for the calculation after the movement of the node i for old community r to the new community s. The term $q' = \frac{m'_{int}}{m'}$, where m'_{int} is the internal number of edges after node i has been moved from community r to new community s, and m' the total number of edges after the movement of the node i into new community. For n nodes the term $\langle q'\rangle = \frac{m'_{int}}{m'}$, where $m = \begin{pmatrix} n \\ 2 \end{pmatrix}$ are the possible ways of drawing m edges after the movement, and $m'_{int} = \begin{pmatrix} n_c \\ 2 \end{pmatrix}$ are possible ways of drawing internal edges.

- **SBM** [11]:

  $$\Delta Q_{\textsc{sbm}} = \textstyle\sum_{t\neq r,s}[a(m_{rt} - K_{it}) - a(m_{rt}) + a(m_{st} + K_{it}) - a(m_{st})] + a(m_{rs} + K_{ir} - K_{is}) - a(m_{rs}) + b[m_{rr} - 2(K_{ir} + u_i)] - b(m_{rr}) + b[m_{ss} + 2(K_{is} + u_i)] - b(m_{ss}) - a(\kappa_r - K_i) + a(K_r) - a(K_s + k_i) + a(K_s).$$

  Where m_{rt} is the number of link between the old community to the rest of the network not including the new community, K_{it} is the number of edges between the nodes i and the nodes in group t without including the self-edges, u_i is the number of the self-edges of the node i, m_{st} is the number of links between the new community and t, K_{ir} is the number of node i inside the old community, K_{is} is the number of links between node i and the new community, m_{ss} is the internal number of links of new community, m_{rr} is the internal number of links of old community, K_r is the total degree of the old community and K_s is the total degree of the new community. $a(p) = 2p\log p$ and $b(p) = p\log p$.

Appendix B

The most widely known objective function for community detection is modularity. As mentioned in [26], modularity finds the best divisions based on the optimization of the function itself. It makes a comparison of the number of links within each community with the expected number of links in a random graph of the same size and same distribution of node degrees, then sums the differences between the expected and observed values for all the communities.

As explained [17], suppose we have a division of the network into two groups r and s. If node belongs to group r then $r_i = 1$ and if node is belong to group s then $s_i = 1$. Then, modularity Q is given by the sum of $A_{ij} - K_i K_j / 2m$ over all pairs of nodes i, j that falls in the same group.

$$Q = \frac{1}{2m} \sum_{i,j} \big(A_{ij} - \frac{K_i K_j}{2m}\big)(r, s)$$

Q is the quality function.

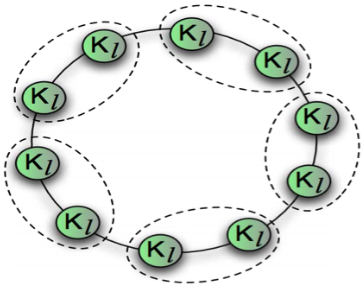


Figure 15: Resolution limit illustration [17]

However, modularity is not consistent, because of the resolution limit problem. It favors network partition with groups of modules combined into larger sub-communities therefore it prevents the detection of smaller communities in larger networks.

In Figure 15 we highlight the problem of resolution limit. The natural community structure of the network, represented by the individual cliques (circles), is not recognized by optimizing modularity if the cliques are smaller than a scale depending on the size of the network. In this case, the maximum modularity corresponds to a partition whose clusters include two or more cliques.[18]

The roots of the problem derive from the sum of each term, because modularity is a sum of terms, where each of them corresponds to a module. Therefore, it might miss important structures of the network. This complication leads to a new challenge, finding the correct objective function that would be focused on the local definition of community, regardless its size.

References

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[2] Rodrigo Aldecoa and Ignacio Marín. Surprise maximization reveals the community structure of complex networks. *Scientific Reports*, 3:1060, 2013.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[4] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[5] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publicated Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[6] Leonhard Euler. Leonhard euler and the königsberg bridges. *Scientific American*, 189(1):66–72, 1953.

[7] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[8] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[9] Santo Fortunato and Claudio Castellano. Community structure in graphs. In *Computational Complexity*, pages 490–512. Springer, 2012.

[10] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[11] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

[12] Bisma S. Khan and Muaz A. Niazi. Network community detection: A review and visual survey. *arXiv:*, 1708.00977, 2017.

[13] Ivan Kojadinovic. On the use of mutual information in data analysis: an overview. In *Proc Int Symp Appl Stochastic Models Data Anal*, pages 738–47, 2005.

[14] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[15] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[16] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[17] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[18] Mark EJ Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, 2016.

[19] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663, 2004.

[20] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 2006.

[21] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[22] Suman Saha and Satya P Ghrera. Network community detection on metric space. *Algorithms*, 8(3):680–696, 2015.

[23] Tiziano Squartini, Francesco Picciolo, Franco Ruzzenenti, Riccardo Basosi, and Diego Garlaschelli. Disentangling spatial and non-spatial effects in real complex networks. In *Proceedings of the International Conference on Complex Networks*, 2014.

[24] Vincent Traag. *Louvain algorithm for igraph.* accessed August 2018, http://louvain-igraph.readthedocs.io/en/latest.

[25] Vincent A Traag, Rodrigo Aldecoa, and J-C Delvenne. Detecting communities using asymptotical surprise. *Physical Review E*, 92(2):022816, 2015.

[26] Vincent A Traag, Paul Van Dooren, and Yurii Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1):016114, 2011.

[27] N.J. van Eck V.A. Traag, L.Waltman. From louvain to leiden, avoiding badly connected communities. 2017.

[28] Duncan J Watts. Networks, dynamics, and the small-world phenomenon. *American Journal of sociology*, 105(2):493–527, 1999.

[29] Zhao Yang, René Algesheimer, and Claudio J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750 EP, 2016.

[30] Pan Zhang. Evaluating accuracy of community detection using the relative normalized mutual information. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(11):P11006, 2015.