Community detection - the Kemeny constant as cluster quality function

Marnix Maas
VU University Amsterdam, The Netherlands
marnixmaas@live.nl

Supervisors:
Daphne van Leeuwen
Elenna Dugundji

October 19, 2018

Abstract

This study aims to propose a cluster quality function for community detection based on a characteristic of Markov-chains, called the Kemeny constant. The viability of the cluster quality function is evaluated using benchmark networks with known solutions. Then, the cluster quality function is applied to a real-world network depicting travel patterns in Amsterdam.

Contents

1 Introduction 4

2 Literature review 5
- 2.1 Community detection . . . 5
- 2.2 Cluster quality functions . . . 5
  - 2.2.1 The Kemeny constant . . . 6
- 2.3 Community detection in real-world networks . . . 6

3 Data description 8
- 3.1 Zachary's karate club . . . 8
- 3.2 Artificial network . . . 8
- 3.3 Amsterdam travel data . . . 10

4 Methods 12
- 4.1 Modularity . . . 12
- 4.2 Louvain algorithm . . . 12
- 4.3 Cluster quality function . . . 13
- 4.4 Partition similarity measure . . . 14

5 Results 16
- 5.1 Zachary benchmark . . . 17
- 5.2 Artificial network benchmark . . . 20
- 5.3 Community detection in Amsterdam . . . 23

6 Conclusion 25

7 Discussion 26

8 Acknowledgements 27

9 Appendix 30
- 9.1 Benchmark graph generation . . . 30
- 9.2 Additional figures . . . 31

1 Introduction

Detecting communities in networks is a subject that has received considerable attention over the last years, as has been noted by Ciglan et al. [17]. The objective of community detection has an ambiguous definition. Still, several techniques and new algorithms have been proposed, but the ambiguity concerning the objective makes assessment of performance difficult. As a result, the ambiquity slows down progress in the field (Fortunato & Hric, 2016, p. 2). Despite the lack of a uniformal definiton of good partitions, community detection has found numerous applications to real world networks [3, 2, 7]. Finding communities (or clusters) in a network may lead to better understanding of the network and allows for a more in-depth look on each individual cluster.

Many algorithms for community detection (or graph clustering) use a measure to describe quality of the partitions. The most popular of them is called modularity. Other measures focus on the quality of the clusters themselves. In this study a new cluster quality function ('CQF') based on the Kemeny constant is evaluated. The Kemeny constant is a measure of connectivity of a Markov-chain [6]. Since a graph can be considered as a Markov-chain, the Kemeny constant may prove to be a suitable CQF. Furthermore, to gain insight in its application to a real-world dataset, the CQF is applied to a network from a study of van Leeuwen et al. [7].

In their study they aim to gain insight in travel patterns in Amsterdam using community detection. Other travel data are usually limited to a single source, for example passengers of a metro system. However, to draw a complete map of travel patterns, all movement within the city should be taken into account. To this end, Google has provided a new data source with location information from Android phones. The data includes travel intensities between 481 neighbourhoods across Amsterdam. The study and data will be further described in section 2 and 3.

This study elaborates on describing quality of clusters with the Kemeny constant. The CQF will be incorporated in the Louvain algorithm [4], a popular method for community detection. Consequently, this paper aims to answer:

- Can the Kemeny constant be used as an appropriate cluster quality function?

To study the research question, benchmark networks will be used. Finally, the method is applied to the Amsterdam travel data.

This paper is organized subsequent sections. Firstly, research into community detection and quality measures will be described in 2, the literature review. Then, the data will be described in more detail in 3, data description. Section 4, methods, will expand on the techniques in this study, as well as the approach to the research question. It is followed by 5, the result of this study. Lastly, a conclusion will be given in 6, which is discussed in 7, the chapter thereafter.

2 Literature review

In this section the problem of community detection and its validation are discussed. Followed by a consideration of literature on cluster quality functions. The proper definition and an intuitive interpretation of the Kemeny constant are shown. Lastly, several papers with an application of community detection to a real-world dataset are presented. These papers serve to gain insight in solutions presented for data similar to the Amsterdam travel data.

2.1 Community detection

Community detection (or graph clustering) is problem in the field of mathematics and computer science, which aims to find groups within a graph (or network). Though, it has also found applications within social sciences and engineering. Figure 1 in section 3 shows an example of a network. The groups are also called clusters or communities, hence the name community detection. Community detection can help to better understand the structure of a graph. Additionally, it supports a more in-depth study of a segment of the network: the individual clusters.

Fortunato has written an extensive overview on community detection in graphs in 2010 [10]. More recently (2016) a brief tour through the problem of community detection and popular methods with their strengths and weaknesses was written by Fortunato and Hric [5]. It should be noted that no clear definition for the problem of network clustering exists. However, Newman and Girvan have proposed modularity as a quality function on the goodness of partitioning of a graph [1]. Modularity is the most popular quality function for algorithms in community detection (Fortunato & Hric, 2016, p. 27). An efficient and well performing method for the problem has been devised by Blondel et al. [4], called the "Louvain algorithm". It is a greedy algorithm which optimizes modularity in each iteration.

"The accuracy of clustering techniques depends on their ability to detect the clusters of networks, whose community structure is known" (Fortunato & Hric, 2016, p. 12). It enables the comparison of the outcome of a clustering technique with the actual clusters. While several methods exist, there is no universal definition for the similarity of partitions. Still, Fortunato and Hric suggest to utilize a measure based on information theory. They describe both 'normalized mutual information' and 'variation of information', respectively NMI and VI.

2.2 Cluster quality functions

Modularity is the most used quality function for optimizing the partition of a network. However, Fortunato and Hric (2016, p. 31) suggest that optimizing a cluster quality function offers several advantages. Firstly, it fits better with the idea that a community is a local structure. Secondly, changes in its structure should not affect the network as a whole, which could prevent resolution limit problems. Methods that suffer from the resolution limit have difficulty with detecting clusters with a (relatively) small number of nodes. Modularity suffers from the resolution limit as well, its usefulness has therefore been questioned by Fortunato and Barthlemy[11].

An application of the Kemeny constant for graph clustering is proposed by Berkhout and Heidergott. In their paper [8] they aim to "elaborate on the Kemeny constant as measure of connectivity of the weighted graph associated to a Markov chain" (2018, p. 1), as well as its application to cluster analysis. The method is applied to four different benchmark networks. They suggest that this new approach to graph clustering seems promising.

2.2.1 The Kemeny constant

The Kemeny constant is an attribute of a Markov-chain. It "denote the mean time from i to equilibrium, meaning the expected time, starting from i, to arrive at a state selected randomly according to the equilibrium measure w of the chain" (Doyle, 2009). Since the value does not depend on the starting state i, the value is a constant. In a study by P. Doyle the definition of the Kemeny constant is given, as well as an example to intuitively explain the constant [6]. The Kemeny constant denotes the connectivity of a Markov-chain, thus a well connected chain should show lower values for the constant. For more material on Markov-chains, see Doyle [6] or Meyn [18].

2.3 Community detection in real-world networks

The studies discussed below have applied community detection on a real-world dataset. These examples serve to gain insight in the application of graph clustering on existing networks.

Earlier research into travel patterns has been carried out by Van Leeuwen et al. [7] in 2018. It consists of the clustering of travel patterns in Amsterdam from both the perspective of space and time. The clusters are evaluated on robustness, spatial connectivity and similarity to regional districts in Amsterdam. Clusters are detected with the Louvain algorithm based on the optimization of modularity. Even though the algorithm is not restrained with any spatial limitations, the clusters found are mostly adjacent. Additionally, they show similarity to the regional districts of Amsterdam.

In similar research, Blondel et al. [3] aim to propose regional boundaries based on communication via mobile phones. In addition, they compare regional districts with the communities based on mobile communication. Consequently, they assess the distance of phone calls within the country. Two analyses are carried out: for the relative frequency of calls and the average duration. For the detection of communities the Louvain algorithm used. Interestingly, the two analyses lead to different results. The latter seems to be affected by the dominantly used language of the areas. Although the study considers (mobile) communication rather than travel data, there is a similar goal to the study by van Leeuwen et al. Also, the study evaluates the robustness of communities by repeating the algorithm on different permutations of the data (robustness of communities will be expanded on in chapter 4.2).

In 2010 Ratti et al. [2] carried out another study on mobile communication data within Great-Brittain. The article aims to compare boundaries defined by the government with clusters found in the communication data. Communities are produced with optimisation of spectral modularity [9]. With an additional constraint that regions must be adjacent. However, they also consider an analysis without the spatial constraint. The article ends on a note that similar analysis could be applied to movement patterns for new perspectives on transportation planning. Which is precisely the aim of finding community structures in the Amsterdam travel data.

To summarize, the most popular method for community detection, modularity, suffers from several limitations. As a result, this study aims to elaborate on the Kemeny constant as a cluster quality function and to show its usefulness for community detection. In addition, its application to a real-world dataset (Amsterdam travel data) is explored.

3 Data description

For this study two benchmark networks as well as the Amsterdam travel data are used. Benchmark networks serve to gain insight in the usefulness of the Kemeny constant. Thereafter, the Kemeny constant is used to illustrate the quality of clusters found in the Amsterdam travel data. This section elaborates on these networks.

3.1 Zachary's karate club

The first benchmark graph is a network called Zachary's karate club, shown in figure 1. It was introduced by Zachary [14] as a study of social relationships. In short, it represent members of a karate club. After a conflict, the club has split into two groups. Since its use by Girvan and Newman in 2002 [16] it has often been used for the purpose of community detection. The connections (edged) between the members are known as well as the result of the division afterwards. Therefore, it is a convenient example for community detection.

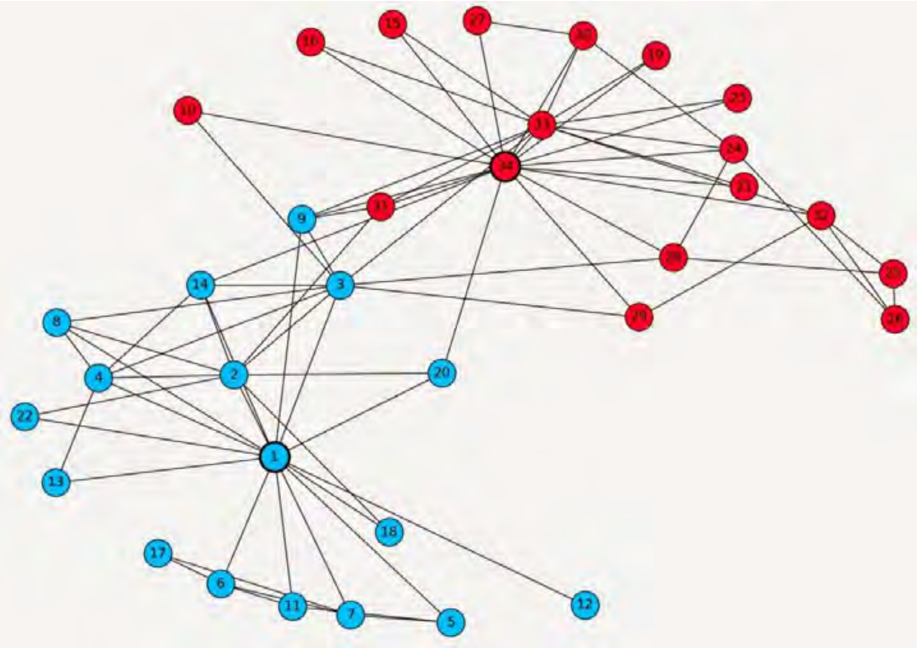


Figure 1: The Zachary karate club network, colors represent the two groups that formed after the conflict (Source: [24]).

3.2 Artificial network

For most real-world networks the community structure is not known (or non-existent), the Zachary network proves a convenient exception. Fortunately, networks can be generated in order to validate community detection techniques. Lancichinetti et al. [19] have recommended that these generated networks should follow a power-law distributions to better represent the characteristics of real networks. In a study by Lancichinetti and Fortunato [20] they used generated networks to asses community detection algorithms. Since the community structure of the generated networks are known, they form a suitable benchmark.

For the current study an artificial network has been generated using the algorithm from a study by Fortunato and Lancichinetti [23]. Fortunato [25] has made the code to generate such networks publicly available. The algorithm accounts for important characteristics of real networks, such as the power-law distribution for node degree and community size. Additionally, the generated networks can include weight and direction. The network to be generated can be adjusted with several parameters. The benchmark graph for this study was generated with standard parameters supplied in the code package. A further description of the graph generation can be found in the appendix 9.1.

In summary, the generated network has 33 known communities, which can be used for the benchmark. The communities have an average size of 30 nodes, with a minimum of 20 and maximum of 49. A histogram of the (non-zero) edge weights is shown in figure 2.

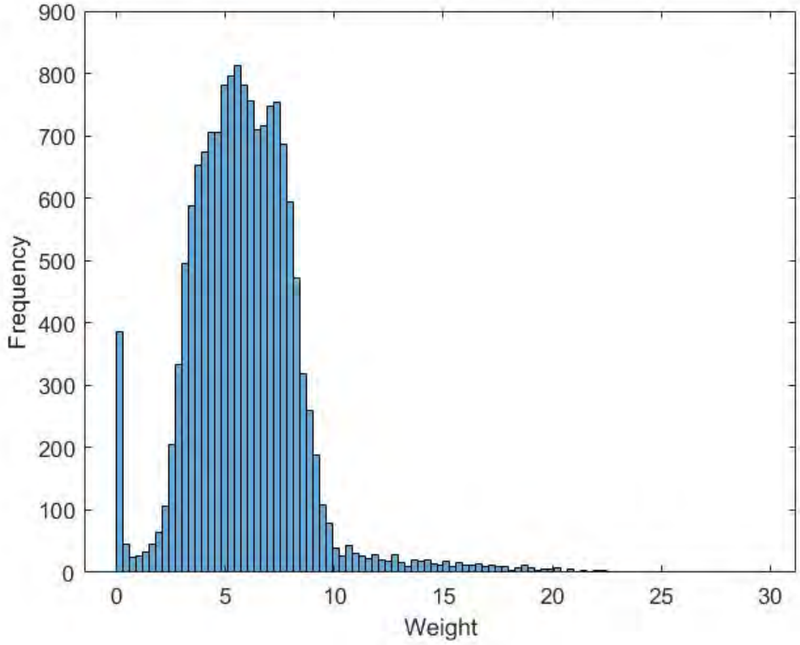


Figure 2: Histogram with non-zero edge weights of the artificial network

3.3 Amsterdam travel data

Google has provided a dataset with location information from Android phones for every hour over a six month period. This 481x481 matrix denotes the amount of people that have traveled between neighborhoods across Amsterdam divided by the maximum number found in the dataset. In other words, it shows the intensity of travel between two neighborhoods. Figure 3 displays a map of Amsterdam and an outline of the neighborhoods. The intensities have been given, rather than the total amount, to ensure anonymity. In addition, the intensities are specified with a direction: origin and destination (OD-pairs). The data can be considered a network, where each neighborhood is a node and the intensities are edges. Similarly, it be described as a Markov-chain, where each travel intensity represents the weight for transitioning between neighborhoods (states).

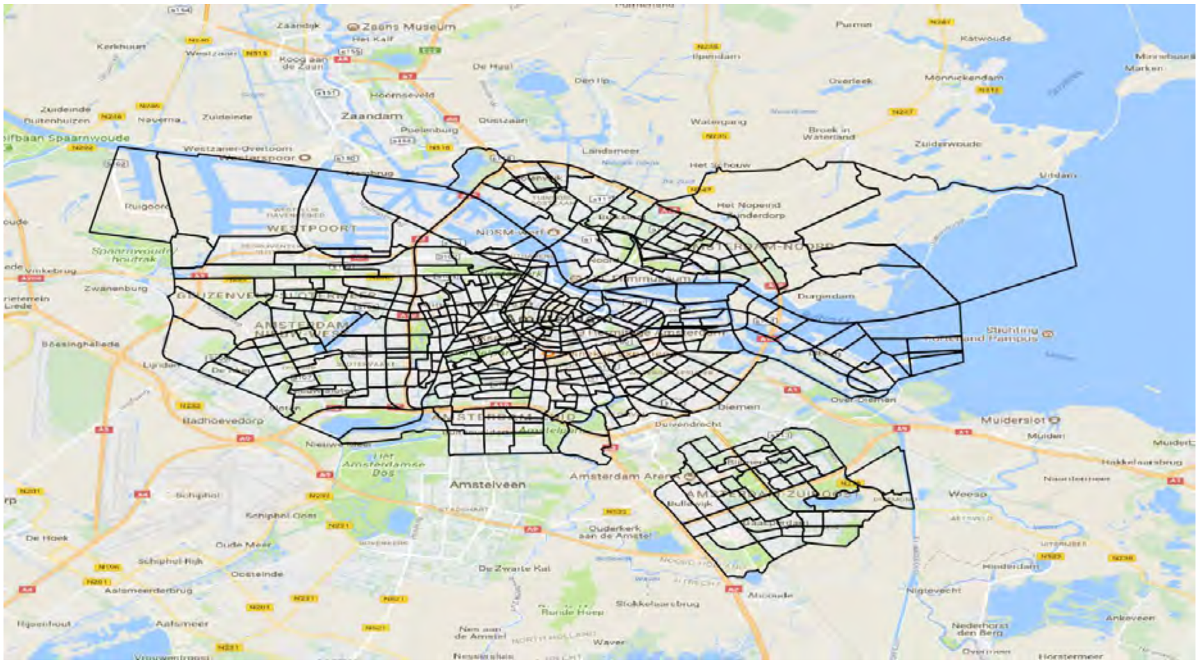
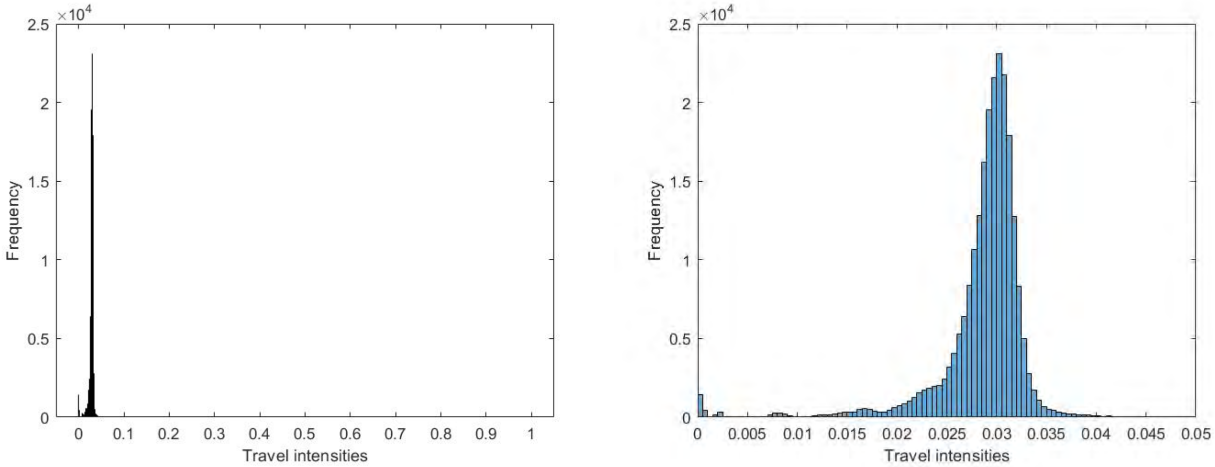


Figure 3: Map of Amsterdam with an outline of the 481 neighborhoods

For the current study, the overall travel behavior is of interest. Therefore, the intensities are aggregated over the whole six month period. Figure 4a shows a histogram of normalized intensities. Most links in the network are relatively small. Hence, figure 4b shows the distribution of all intensities below 0.05.

As mentioned by Ratti et al. [2] analyzing movement patterns offers potential for improving urban infrastructure and planning. Those analysis can be assisted by the community structures. In addition, it allows for a more in-depth study of the individual clusters.



(a) all intensities

(b) intensities below 0.05

Figure 4: Histograms of normalized travel intensities

4 Methods

This chapter will elaborate on the techniques used for community detection. In addition, the new proposed cluster quality function using the Kemeny constant will be discussed. Lastly, the approach to evaluating the usefulness of the CQF is expanded on.

4.1 Modularity

Referring back to Fortunato & Hric, for community detection methods based on optimization of a quality function, modularity is the most popular. It measures the quality of partitions in a graph. The partition quality function was devised by Newman and Girvan in 2003 [1]. Some intuition is given by Ratti et al. (2010, p. 2): "High modularity values occur when the network is subdivided such that there are many links within communities and few between them, as compared to a randomly generated network with otherwise similar characteristics.". The definition of modularity is given by:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{K_i K_j}{2m} \right) \delta(C_i, C_l) \tag{1}$$

where m is the number of edges, A_{ij} is the element of the adjacency matrix, K_i and K_j denote the total weight of edges connected to nodes i and j respectively, and δ indicates whether the nodes belong to the same community (1 if they belong to the same community, 0 if they do not). The maximization of modularity is NP-hard [12]. However, gain in modularity is efficiently calculated when modifying the partition of a network. This property is an advantage in heuristics, such as the Louvain algorithm, as it decreases computation time.

4.2 Louvain algorithm

Numerous algorithms exist for the optimization of modularity. Since the problem is NP-hard [12], heuristics are needed to retrieve a solution in feasible time. A well performing and efficient algorithm often used for community detection is the Louvain algorithm [4]. This study will not expand on the (dis)advantages of using this particular algorithms, nor on its competitors. The Louvain algorithm divides a graph in partitions in a greedy manner through (one or several) iterations. In the study by Blondel et al. the algorithm is described thoroughly [4]. A brief explanation on the steps of the algorithm is given underneath:

Initially, all nodes in the network are considered as a separate community.

1. For a random permutation, loop over all nodes. Calculate gain in modularity when removing the node from its current community and adding it to the community of each of its neighbors. The node will be moved to the community that results in highest modularity gain, if that gain is positive. Ties in

modularity gain are solved by arbitrary assignment. Once all nodes have been considered the first partition of the network is created.

2. For the second iteration, each clusters is considered (combined) as a single node. Repeat the first step on this smaller network. In this step communities will be merged so long as it results in a higher modularity. Step two is repeated until communities between two steps are equal: the partition converged.

The partitions are established in hierarchical nature due to (repetition of) the second step. It has been suggested that "using the lowest level helps avoiding unnatural community mergers" (Fortunato & Hric, 2010, p. 30). Unfortunately, it is unknown in what situations the performance improves if subsequent steps are carried out. On the other hand, it does leave the option evaluate the outcome in a more intuitive manner. Using "concensus clustering" more a robust partition could be derived. Zhang and Moore [13] have shown that "the consensus of many high-modularity partitions, combined with a hierarchical approach, could help to resolve resolution problems and to avoid to find communities in random graphs without groups" (Fortunato & Hric, 2016, p. 31). Moreover, the Louvain algorithm does not require the user to provide the number of clusters beforehand. This is especially useful since the number of clusters is often unknown. As is the case for the Amsterdam travel data.

Lastly, the concept of robustness is considered. Each run of the Louvain algorithm may discover different partitions. Thus, one may examine whether nodes are included in the same cluster each run. If so, the node (or cluster) is considered robust, since the random permutations of the algorithm does not influence the clustering.

4.3 Cluster quality function

Modularity is the most used clustering technique for optimizing a partition quality function. Instead, Fortunato and Hric suggest that optimizing a cluster quality function offers several advantages. Therefore, this study aims to evaluate the usefulness of the Kemeny constant as a cluster quality function. This study does not aim to compare the Kemeny constant with other cluster quality functions, but rather to show whether it is viable.

The Kemeny constant is a charactaristic of a Markov-chain. Consider a Markov-chain and let the expected time to go from state i to state j be M_{ij} and let $M_{ii} = 0$. The average time to reach the equilibrium from state i is

$$M_{iw} = \sum_j M_{ij} w^j$$

where w is the equilibrium distribution and w^j is the probability that the chain in equilibrium is in state j. John Kemeny found that M_{iw} is independent of i, therefore the value is equal for all initial states. Thus, it is a constant: The Kemeny constant. It can be calculated by taken the trace of the "fundamental matrix". A more detailed explaination of how the Kemeny constant is calculated and an intuitive example for the Kemeny constant are presented in a study by Doyle [6].

In short, the Kemeny constant denotes the expected amount of steps needed to go from a random starting state to a random destination in the equilibrium of a Markov-chain. Therefore, it is expected that a well connected cluster shows small values of the Kemeny constant. That is, in comparison to the Kemeny constant of that cluster with 'uniform' edges. More specifically, in the uniform cluster, the probability of moving to a certain node is the same from each other node in the cluster. This uniform cluster is used as comparison to the Kemeny constant. If the Kemeny constant of a network is smaller than that of the uniform network, it is better connected. It is assumed that the quality of the cluster is higher in this situation.

The uniform cluster serves a second purpose, since the Kemeny constant is expected to increase progressively the more nodes the cluster contains. Due to the larger size of the cluster, the probability of reaching any of the nodes in equilibrium in the cluster is progressively smaller. Despite a high value of Kemeny, the cluster may still be of good quality. But, the Kemeny constant of the uniform cluster will increase accordingly. Thus, dividing the Kemeny constant of the cluster with the Kemeny constant of its uniform cluster results in a measure that ought to be unaffected by size (this will be denoted as "relative Kemeny"). As a result, the relative Kemeny can be compared across clusters of different size. The quality of the complete partition of the network is then calculated by averaging the relative Kemeny over all clusters ("mean relative Kemeny").

To illustrate, should a cluster be of good quality it is expected that it is better connected than its uniform version. Therefore, the Kemeny constant of the cluster should be lower than that of the uniform cluster. As a result the relative Kemeny value will be lower than one. Note that the relative Kemeny can still reach values higher than one if the connectivity of the cluster is worse than the uniform version. Evidently, the quality of the cluster is poor in this case.

The relative Kemeny will be incorporated in the Louvain algorithm to evaluate its relation to the quality of clusters. The benefit of the Kemeny constant as CQF will show if low values for relative Kemeny co-occur with high values of NMI.

4.4 Partition similarity measure

Quality of partitions can be measured in numerous ways. However, it was recommended by Fortunato & Hric (2016, p. 15) to use a measure based on information theory. The measurs serve as an estimate for the similarity of two partitions of a network. While Fortunato and Hric describe both NMI and VI in their study, they note that VI "is a more promising measure". However, emphasize is put onto the fact that the measure does not perform well when the partitions become very dissimilar. Additionally, they note that "it shows unintuitive behavior in particular instances (Delling et al., 2006).". Therefore, this study uses NMI to calculate partition similarity.

Normalized mutual information (NMI) can also estimate the accuracy of a partition when compared to the true partition. NMI has been regularly used for this purpose since a comparative analysis was carried out by Danon et al. [15]. Let X be a vector containing partition labels for comparison and Y the true labels. Then, NMI is defined as:

$$NMI = \frac{2 * I(X,Y)}{H(X) + H(Y)}$$

Where I(X,Y) is the mutual information. H(X) and H(Y) are defined as the Shannon entropy of vector X and Y respectively. Due to the normalization, NMI only takes on values between 0 and 1. If the partitions are the same, NMI is 1. If the partitions are completely dissimilar, NMI will be 0. NMI is expected to show high values when the quality of clusters is high, because high cluster quality should imply good partitions.

To summarize, research is done on the effects of using a cluster quality function, called the Kemeny constant. The Kemeny constant of a segment of a network shows how well a cluster is connected to each of its nodes. The better the connections in the community, the lower the expected number of steps between nodes should be. With it, the Kemeny constant of a well connected cluster is expected to be small. Relative Kemeny as CQF is assessed by incorporating it in the Louvain algorithm. Its performance is evaluated using a similarity measure from information theory: NMI.

5 Results

In this section the viability of using the Kemeny constant as a cluster quality function will be examined. This is done by measuring the quality of the relative Kemeny by comparison to NMI.

Since the Kemeny constant is a measure of connectivity of a cluster, it is expected to decrease as the quality of clusters increases. Before using the constant as cluster quality function it should be shown that the above expectation holds. Note that the algorithm is a heuristic based on the optimisation of modularity. Therefore, there is no guarantee that the solution found is the optimal partition of the network nor the hypothesized optimal relative Kemeny.

The quality of clusters is assessed with the NMI, where the partitions are compared to the known solution of the network. Specifically, for each step in the Louvain algorithm (see section 4.2) the NMI and mean relative Kemeny are calculated. Because the Louvain algorithm usually converges fast, few steps will have been taken by the method. Note that Louvain starts at random nodes for each step, therefore partitions can vary between runs. Thus, repeatedly applying the algorithm results in more data points for the comparison.

5.1 Zachary benchmark

Firstly, the well-known network of Zachary's karate club is considered. For this specific network, Louvain often converges in only two or three steps. The algorithm is repeated 500 times, which results in two vectors of approximately 1200 elements. The result is shown in figure 5, a least squares line is included. The mean relative Kemeny value for the known solution is 0.5846.

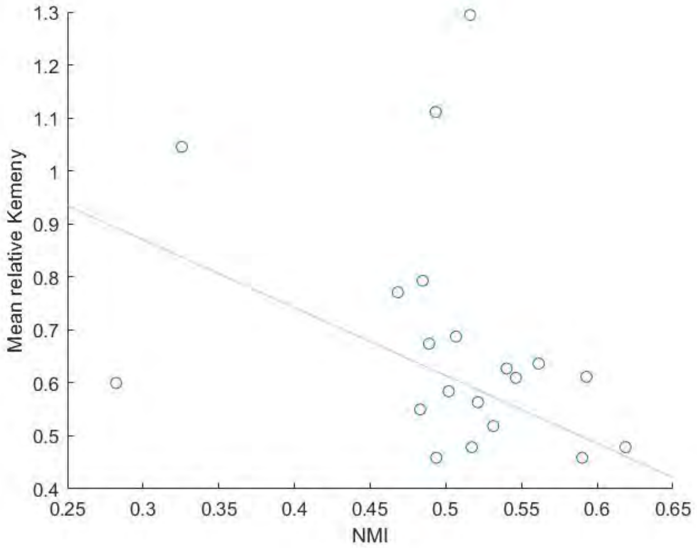


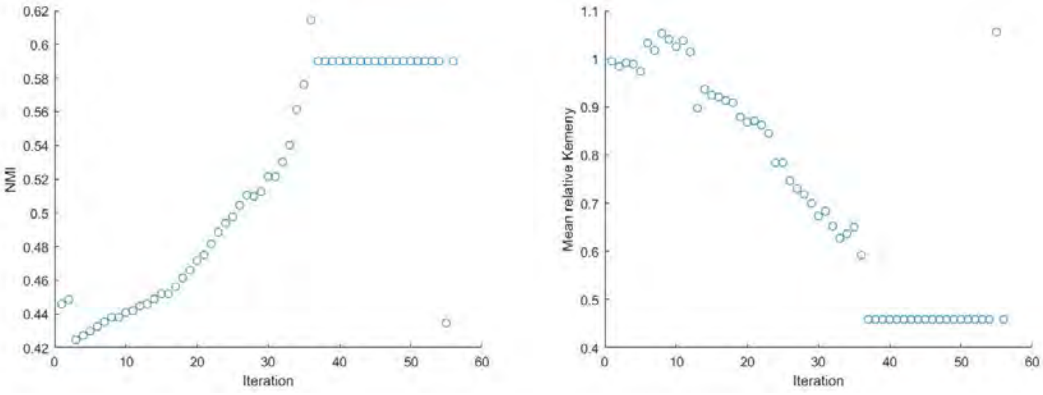
Figure 5: NMI vs mean relative Kemeny on the Zachary network

Surprisingly, despite the number of elements only 20 points are visible in the figure. Due to the small size of the network the Louvain algorithm appears to converge to a handful of solutions in each step. Still, the figure shows a downward trend: as Kemeny decreases the quality of clusters (NMI) increases. Correlation between the vectors is -0.4825, with a corresponding p-value of almost zero $(1.1918e^{-76})$. In conclusion, a negative linear dependence between Kemeny and NMI is shown for this network.

The optimal solution has a mean relative Kemeny of 0.5846. In the figure it can be seen that NMI values range from 0.48 to 0.62 for partition with lower relative Kemeny values. The mean relative Kemeny for the optimal solution was expected to show the lowest value, this does not hold for the Zachary network. In addition, several outliers can be seen. Three points shows a relative Kemeny higher than one and a single point shows a particularly small NMI value. Furthermore, the Louvain algorithm converges to four clusters for this network, although the actual partition has two. Hence, the highest NMI value reached is not particularly high.

Figure 5 is held back by the small number of possible outcomes for each step in Louvain. Also, four outliers are shown that cannot be justified. Perhaps more information can be gained by examining each iteration of Louvain, rather than just the end of each step. Similarly to the expectation of the steps of the algorithm, each iteration of Louvain is expected to come closer to the actual partitions of the network. The Kemeny constant of each iteration is then supposed to decrease as well.

The sequence of NMI and Kemeny values for each iteration of a single run of the Louvain algorithm are shown in figure 6a and 6b. Generally, each iteration results in a higher NMI and lower mean relative Kemeny. However, the NMI sequence does show two outliers, one where the NMI is higher than iterations afterwards (iteration 36) and one shows a sudden drop in NMI (iteration 55). Although iteration 36 does not show the lowest mean relative Kemeny value, it is closest to that of the known solution (0.5846). Interestingly, the same iteration corresponds to the highest NMI value found. Iteration 55 has a particularly high mean relative Kemeny value, but it also sharply decreases the NMI.



(a) NMI sequence

(b) Mean relative Kemeny sequence

Figure 6: Sequence of NMI and mean relative Kemeny values for a single run on the Zachary network

Figure 7 displays those values compared to each other. A clear (linear) dependence is shown. Indeed, studying the relation between NMI and mean relative Kemeny for iterations of a single run seems to comply with earlier findings.

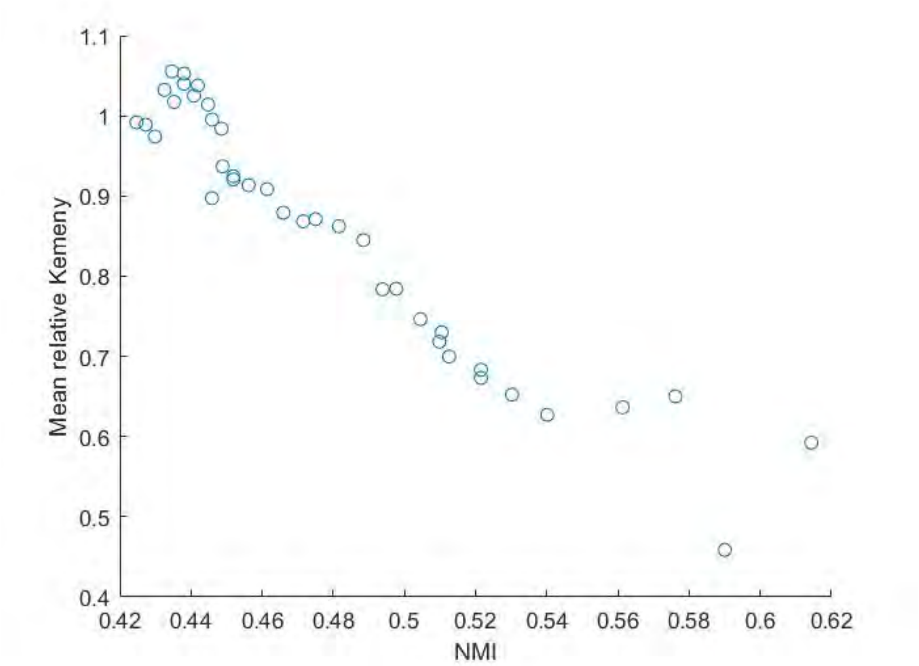


Figure 7: NMI vs mean relative Kemeny in each iteration of a single run on the Zachary network

Moreover, the sequence from multiple runs of Louvain can be combined to show an expanded picture of the relation. For 15 runs of Louvain, the results are shown in figure 8. Although the relation might not look as strong as the run presented in 7, a downward trend can be seen. Correlation of NMI with the mean relative Kemeny is -0.5461, with a p-value of near zero ($8.7137e^{-72}$). NMI has a range of 0.33 to 0.65 for lower mean relative Kemeny values than the known solution.

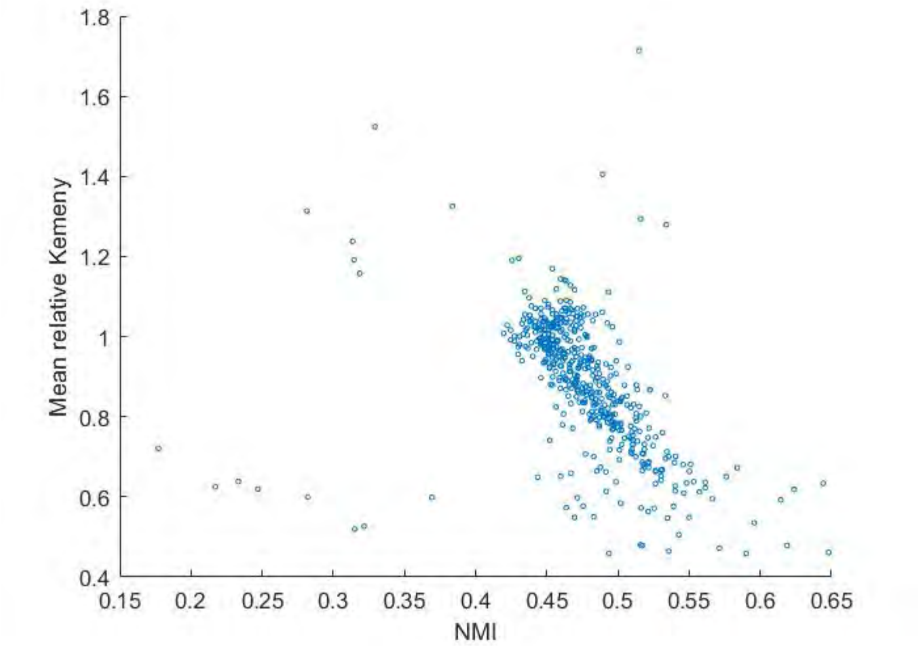
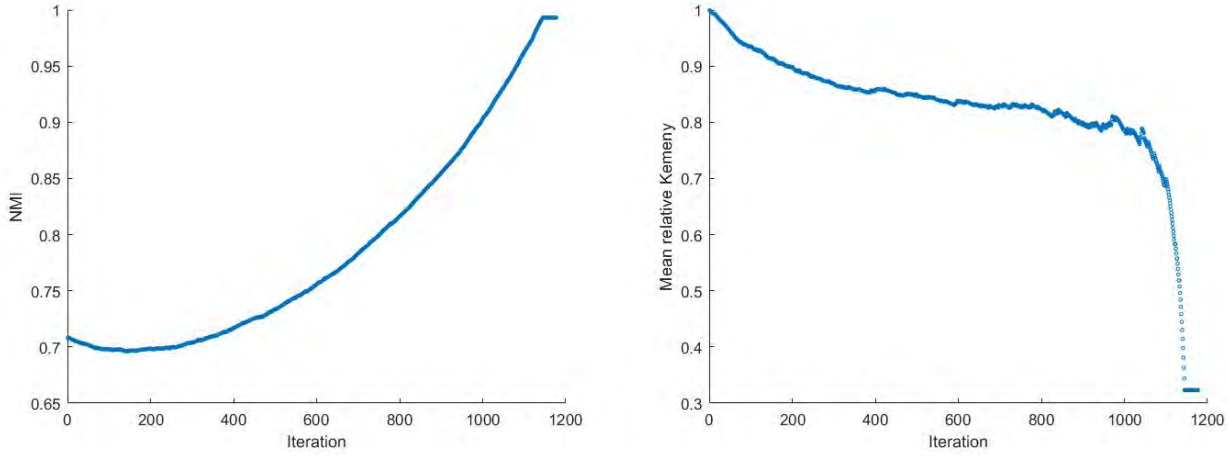


Figure 8: NMI vs mean relative Kemeny for 15 runs on the Zachary network

5.2 Artificial network benchmark

An artificial benchmark graph has been generated to investigate in addition to the Zachary network. This generated network of 1000 nodes consists of 33 communities. The mean relative Kemeny value for the known solution is 0.3163. To calculate the relative Kemeny values a transformation on the network was carried out, see the discussion (section 7) for more details.

The sequence of NMI and mean relative Kemeny values for each iteration of a single run of the Louvain algorithm are shown in figure 9a and 9b. The NMI value decreases until approximately 200 iterations. Thereafter, the NMI values shows a steady increase until the optimal value is approached. It seems that the optimization of modularity result in a near optimal solution for this run. Most notable in figure 9b is that the mean relative Kemeny shows a rapid decrease when NMI approaches the optimal value. The smallest mean relative Kemeny in the figure has a value of 0.3163, which is the same as for the known solution. The NMI and mean relative Kemeny values are displayed against each other in figure 10. Apart from the left tail of the graph, the mean relative Kemeny and NMI appear to have a negative relation.



(a) NMI sequence

(b) Mean relative Kemeny sequence

Figure 9: Sequence of NMI and mean relative Kemeny values for a single run on the artificial network

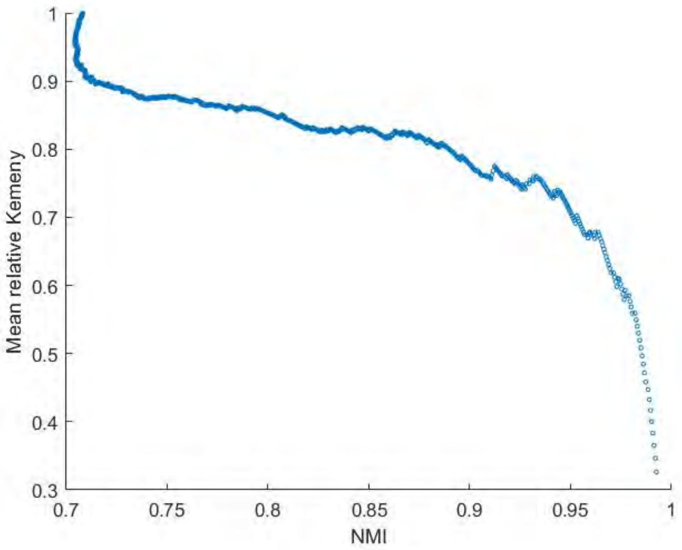


Figure 10: NMI vs mean relative Kemeny values for a single run on the artificial network

To evaluate the relation for multiple runs of the algorithm, figure 11 is displayed. The figure contains five runs, to be able to distinguish between the different runs. The overall behavior of the runs seem to be similar. Note that one realization of the algorithm reaches the optimal solution, therefore the NMI value reaches 1. Even though the figure does not seems to show the same (linear) relation as the result for the Zachary network, a negative relation is displayed. Still, a correlation of -0.8348 was calculated, with a p-value of 0. Keep in mind that the figure is skewed due to the fast decrease of mean relative Kemeny once the NMI approaches 1. Again, the negative relation between the NMI and mean relative Kemeny is supported.

Figure 16 in the Appendix displays the relation with NMI values below 0.95. In this figure the linear relation is better expressed visually, hence the correlation value.

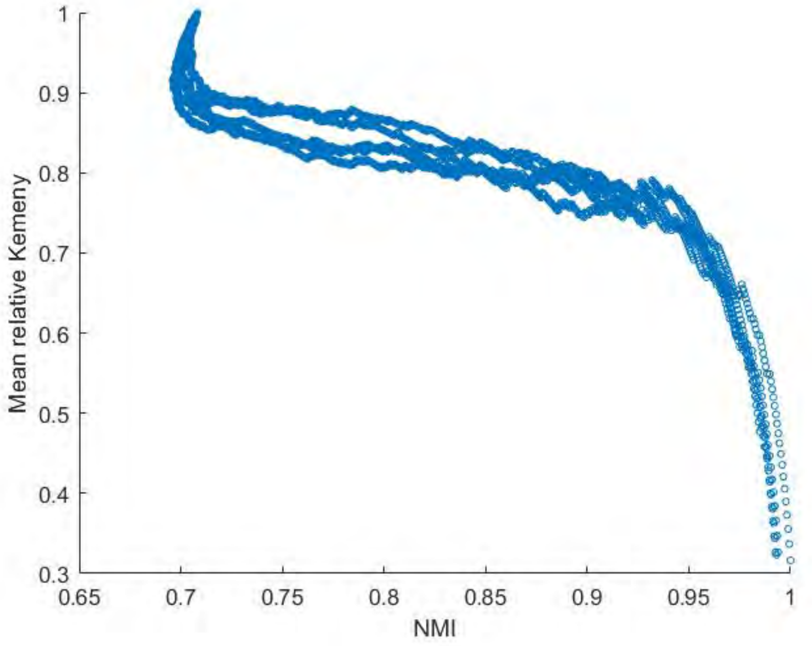


Figure 11: NMI vs mean relative Kemeny for 5 runs on the artificial network

To summarize, comparing NMI and relative Kemeny for each step in the algorithm shows first evidence of the negative relation. To increase the amount of observations the same relation for each iteration of Louvain is considered. As a result the linear dependence between NMI and mean relative Kemeny is confirmed for the Zachary network as well as the artificial network. However, Louvain yields mean relative Kemeny values that are lower than of its known solution. In case of the artificial network, Louvain reaches (or approaches) the optimal solution. As a result it also produces the same mean relative Kemeny value of the known solution.

5.3 Community detection in Amsterdam

Lastly, the Amsterdam travel network is considered. For this network the Louvain algorithm retrieves either 8 or 9 communities. The Louvain algorithm is applied to the the network, for each iteration of the algorithm the mean relative Kemeny and NMI value are calculated. The sequence of mean relative Kemeny values for an arbitrary[1] single run of the Louvain algorithm on the data is shown in figure 12. Most notable is the spread in mean relative Kemeny values after 600 iterations. In contrast to the benchmark networks, where such spread in mean relative Kemeny values was not show . Furthermore, the mean relative Kemeny reached in the last iteration is 0.8903, whereas the lowest value in the sequence is 0.8419. Still, both values are higher in comparison to values shown for the benchmark networks.

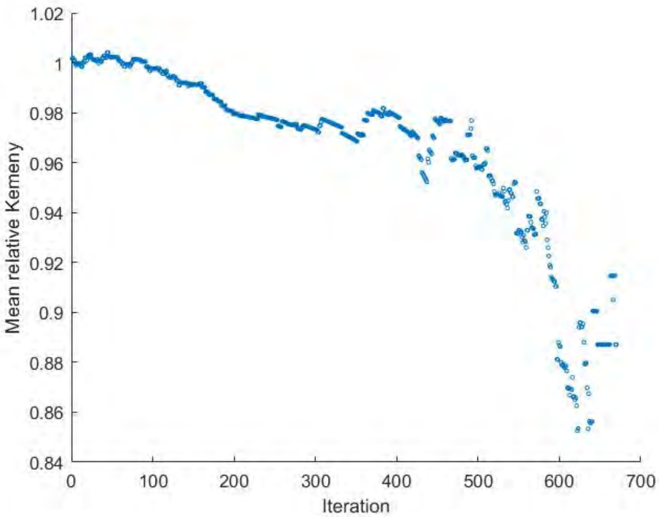


Figure 12: Sequence of mean relative Kemeny values for a single run on the Amsterdam travel network

To retrieve more robust clusters the algorithm is repeated 20 times. Afterwards, each neighborhood is assigned to the community they belonged to most often over the 20 runs. Table 1 displays the relative Kemeny and size of each cluster in the final partition. Several clusters display a relative Kemeny of nearly 1. Therefore, the quality of these clusters ought to be low. However, certain clusters do display lower relative Kemeny values. The three clusters with lowest relative Kemeny seem to have a smaller cluster size. Their cluster quality may be reasonable. For a more intuitive interpretation, the final partition is shown on a map of Amsterdam in figure 13. The cluster with highest quality according to the relative Kemeny is the dark blue cluster located in and around Westpoort and the northern area of Nieuw-West. Cluster 7 in orange shows the highest relative Kemeny value (in hypothesis

[1] seed = 1

the lowest cluster quality). Furthermore, the figure shows that the majority of the neighborhoods are adjacent to a neighborhood of their own cluster, even thought the spatial constraints are not part of the algorithm. Interestingly, the cluster with highest relative Kemeny seems the most divided by another cluster. One clear exception in adjacency is a purple neighborhood located on the right side of the figure. However, the area is completely located on top of IJmeer.

Table 1: Final partition of the Amsterdam travel network

Community	Relative Kemeny	Cluster size (# nodes)
1	0.9870	245
2	0.7086	13
3	0.8506	64
4	0.7584	11
5	0.8812	44
6	0.9683	43
7	0.9960	36
8	0.8233	7
9	0.9923	18

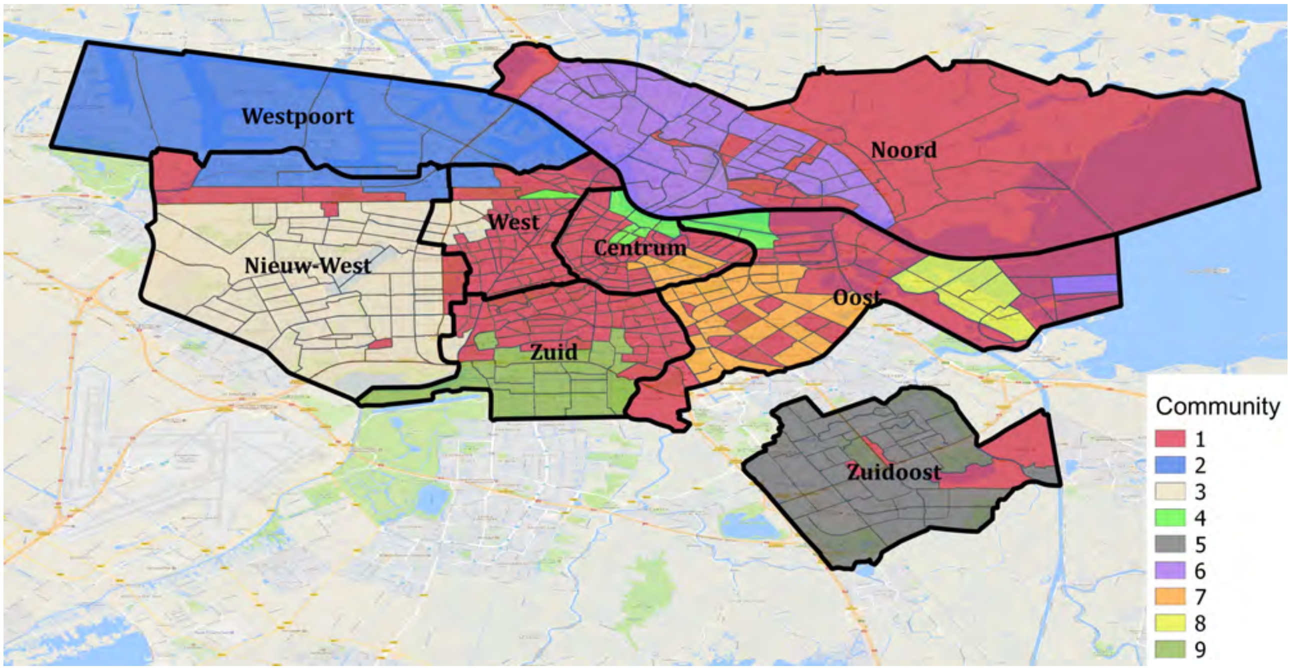


Figure 13: Map of Amsterdam with communities found by the Louvain algorithm

6 Conclusion

The method discussed in this study is an attempt to advance community detection algorithms. Many approaches to the problem have been proposed and some techniques, such as modularity and the Louvain algorithm, have become popularly used methods. Still, these methods may be improved. One suggestion is to use a cluster quality function rather than a partition quality function. To this end, the Kemeny constant is proposed as a CQF for the optimization of community detection algorithms.

The relation between the new CQF (mean relative Kemeny) and the quality of the partition has been evaluated for two benchmark networks. One network denotes a real-world graph and one artificial network was generated following proposed methods. Since the Kemeny constant denotes the connectivity of a cluster, it is expected to show smaller values for a cluster of good quality. Should the relative Kemeny show a negative relation to NMI, it may indeed be a viable CQF.

In the first benchmark graph a (linear) dependence between the quantities was found (figure 7 and 8). A correlation of -0.55 was calculated, with a p-value of near zero ($8.7e^{-72}$). However, multiple outliers were encountered as well. Furthermore, the mean relative Kemeny value for the known solution was higher than several values retrieved by the Louvain algorithm. Most notable in the second artificial network was a steep decrease in mean relative Kemeny values once the partition approached the optimal solution (figure 10 , 11 and 16). In addition, in the first approximate 200 iterations of the algorithm the NMI value as well as the mean relative Kemeny decreased. Still, the linear dependence between the quantities was confirmed with a correlation value of -0.83 and a corresponding p-value of 0. For this network, the optimal solution as well as the mean relative Kemeny value of the known solution were reached. Therefore, from these two benchmark networks, it can be concluded that the relative Kemeny decreases as the quality of clusters increases. However, optimizing the mean relative Kemeny value may not result in a partition with the highest NMI as seen with the Zachary network. To conclude, mean relative Kemeny may still prove a viable CQF. Only if the CQF is introduced in an algorithm and compared to other methods can its viability be determined.

Lastly, the clusters found with the Louvain algorithm in the Amsterdam travel network were evaluated with relative Kemeny. Results are shown in table 1 and figure 13. While several clusters show a relative Kemeny of nearly 1 (meaning that the quality is hardly better than in a uniform cluster), others seem to display higher quality. The lowest value of relative Kemeny for a community is approximately 0.7 in the Amsterdam network.

7 Discussion

This study aimed to show the viability of the Kemeny constant as a cluster quality function. Using the Kemeny constant for graph clustering was also done by Berkhout and Heidergott [8]. Naturally, it is suggested that further research into the CQF involves using it as quality function for optimization in an algorithm. The capability of the Kemeny constant could than be assessed in a similar fashion to studies by Lancichinetti and Fortunato [20], Yang et al. [21] or Emmons et al. [22]. Additionally, the comparison to existing methods may be evaluated. It should be noted that the computational costs of calculating the mean relative Kemeny is higher than calculating the gain in modularity. Therefore, it is expected that optimization with relative Kemeny results in a slower algorithm, unless the method converges considerably faster.

For the Zachary network the Louvain algorithm found partitions with mean relative Kemeny values lower than that of the known solution. Therefore, the optimisation of mean relative Kemeny may be flawed. As the highest quality partition should occur with the lowest mean relative Kemeny value. Nevertheless, it may be that the method of combining relative Kemeny values from all clusters in a network can be improved on. The total score of the network may be influenced by a single bad community. In other words, the overall quality of the partition may improve, but the mean relative Kemeny value is unable to show this. Possibly, taking the median or a weighted average of relative Kemeny values can improve the CQF.

Several outliers of the kemeny constant have been shown in figure 8. These outliers have not been explained and they may influence an algorithm that uses relative Kemeny as optimization metric. In addition, in figure 9a and 9b the mean relative Kemeny value decreases during the first 200 iterations, while the NMI value decreases as well. Since the quality of the cluster should have become worse during these iterations, the Kemeny constant should have increased instead. This may have a negative impact when using the Kemeny constant as CQF.

As mentioned in section 5.2 a transformation on the artificial network was done to calculate the relative Kemeny values. Since the network is very sparse (mostly edges of weight zero, or non-existent), the values for the Kemeny constant became enormous. As a result, calculating the Kemeny constant became troublesome. This was dealt with by adding 1 to all edges. The Kemeny constant could then maintain reasonable values. Consequently, the Louvain algorithm could execute once more. The transformation should not have considerable effect on the result, as all edges are increased identically. The same problem has not occurred for the Amsterdam dataset because it is far from sparse. Only a single edge is missing (weight zero) in the network.

The average modularity of the partitions created by the Louvain algorithm on the Amsterdam Travel data is 0.0098. This is considerably low in comparison to the values reached for other real-world networks, for example Ratti et al. [2] with a modularity of 0.58. Therefore, the quality of the partition found with Louvain is questionable. As a result, the high values for the Kemeny constant may be explained, since the quality of clusters should be low on account of the potential bad quality of the partition.

Figures 6a & 6b and 9a & 9b are combined in figure 14 and 15, respectively, in the Appendix (section 9.2). The mean relative Kemeny has been inverted, so that it may be compared more easily to NMI. These figures may show the relation more intuitively.

8 Acknowledgements

I would like to thank Daphne van Leeuwen for introducing me to the field of community detection and a great deal of support during this paper. In addition, many useful discussions led to this outcome. I would also like to thank Elenna Dugunji for support and suggestions concerning the structure of writing the paper.

References

[1] M. E. J. Newman and M. Girvan (2003) Finding and evaluating community structure in networks.

[2] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton and S. H. Strogatz (2010) Redrawing the Map of Great Britain from a Network of Human Interactions. Page 1, 2, 5

[3] V. Blondel, G. Krings and I. Thomas (2010) Regions and borders of mobile telephony in Belgium and in the Brussels metropolitan zone. Page 1, 3, 4, 9

[4] V. D. Blondel, J. Guillaume, R. Lambiotte and E. Lefebvre (2008) Fast unfolding of communities in large networks

[5] S. Fortunato and D. Hric (2016) Community detection in networks: A user guide. Page 1-2, 9-17, 19, 27-31, 37-38

[6] P. Doyle (2009) The Kemeny constant of a Markov chain

[7] D. v. Leeuwen, J. Bosman and E. Dugundji (2018) Spatio-Temporal Clustering of Time-Dependent Origin-Destination Electronic Trace Data

[8] *Working paper* (2018) J. Berkhout and B. F. Heidergott. Analysis of Markov Influence Graphs. Page 1, 19-24

[9] M. E. J. Newman (2003) Fast algorithm for detecting community structure in networks

[10] S. Fortunato (2010) Community detection in graphs. Page 1-4, 38-41

[11] S. Fortunato and M. Barthlemy (2007) Resolution limit in community detection. Page 6

[12] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner (2006) Maximizing Modularity is hard

[13] P. Zhang, C. Moore and M. E. J. Newman (2014) Scalable detection of statistically significant communities and hierarchies, using message passing for modularity.

[14] W. Zachary (1977) An Information Flow Model for Conflict and Fission in Small Groups

[15] L. Danon, A. Diaz-Guilera, J. Duch and A. Arenas (2005) Comparing community structure identification

[16] M. Girvan and M. Newman (2002) Community structure in social and biological networks

[17] M. Ciglan, M. Laclavik and K. Norvag (2013) On Community Detection in Real-World Networks and the Importance of Degree Assortativity. Page 1

[18] S. Meyn and R. Tweedie (2005) Markov chains and stochastic stability (Rewritten version of Springer-Verlag, 1993)

[19] A. Lancichinetti, S. Fortunato and F. Radicchi (2008) Benchmark graphs for testing community detection algorithms

[20] A. Lancichinetti and S. Fortunato (2009) Community detection algorithms: a comparative analysis

[21] Z. Yang, R. Algesheimer and C. J. Tessone (2016) A Comparative Analysis of Community Detection Algorithms on Artificial Networks

[22] S. Emmons, S. Kobourov, M. Gallant and K. Brner (2016) Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale

[23] A. Lancichinetti and S. Fortunato (2009) Directed, weighted and overlapping benchmark graphs for community detection algorithms

[24] http://historicaldataninjas.com/karate-club-network/

[25] https://sites.google.com/site/santofortunato/inthepress2

9 Appendix

9.1 Benchmark graph generation

An algorithm from Lancichinetti and Fortunato [23] for creating directed and weighted benchmark graphs was used. The algorithm can be provided with the following parameters:

– N: number of nodes
– k: average degree
– maxk: maximum degree
– mut: mixing parameter for the topology
– muw: mixing parameter for the weights
– beta: exponent for the weight distribution
– t1: minus exponent for the degree sequence
– t2: minus exponent for the community size distribution
– minc: minimum for the community sizes
– maxc: maximum for the community sizes
– on: number of overlapping nodes
– om: number of memberships of the overlapping nodes

For the current study an artificial network has been generated with parameters:

```
./benchmark -N 1000 -k 15 -maxk 50 -muw 0.1 -minc 20 -maxc 50
```

The algorithm is requested to create a directed network with a 1000 nodes. In addition, an average node degree of 15 was provided, with a maximum degree of 50. The community size is restricted to 20-50 nodes. Lastly, the mixing parameters for the weight is 0.1.

9.2 Additional figures

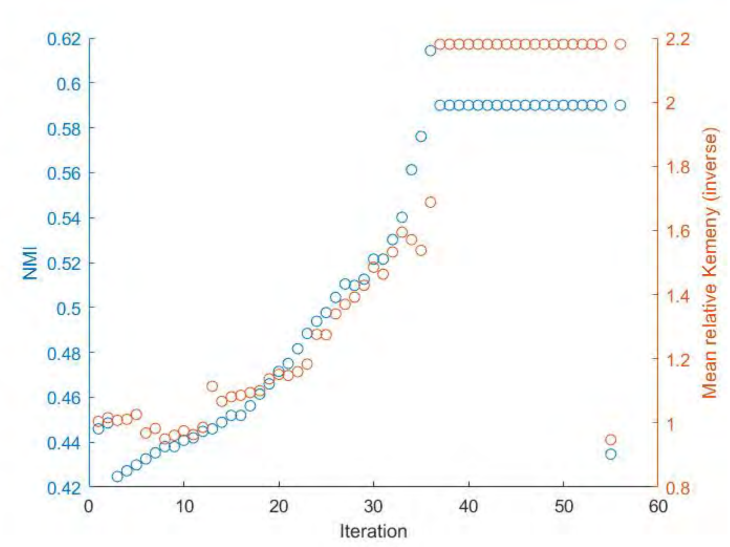


Figure 14: NMI vs inverse of mean relative Kemeny for one run on the Zachary network

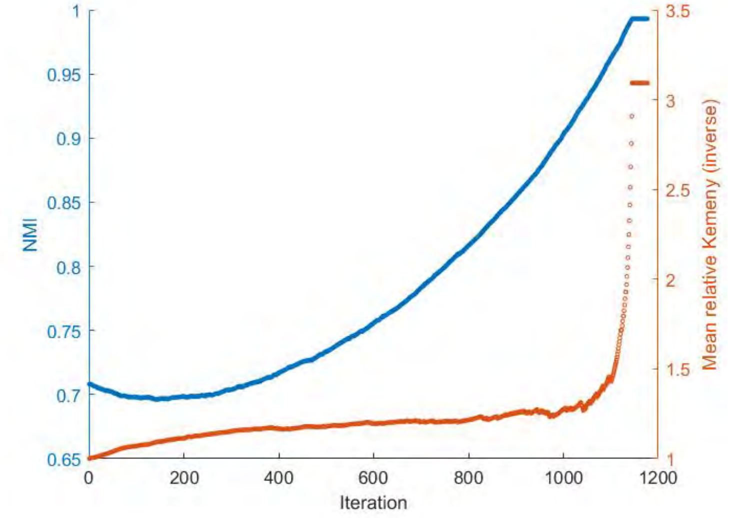


Figure 15: NMI vs inverse of mean relative Kemeny for one run on the artificial network

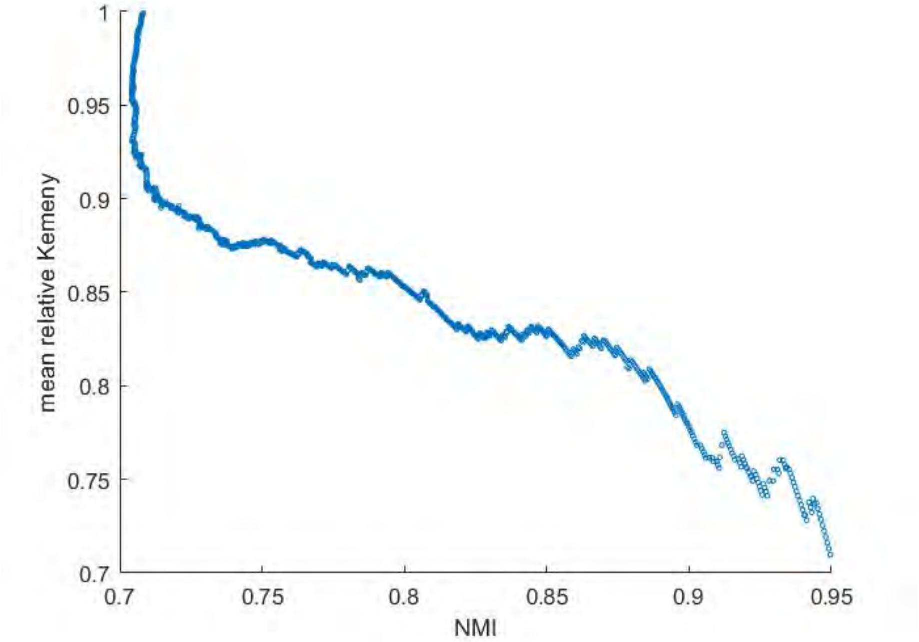


Figure 16: NMI (below 0.95) vs mean relative Kemeny values for a single run on the artificial network