Vrije Universiteit Amsterdam

Research Paper Business Analytics

Optimizing Community Detection Using the Kemeny Constant

Author:
Jiri Brummer
2586530
j.j.brummer@student.vu.nl

Supervisors:
Elenna Dugundji
e.r.dugundji@cwi.nl

Daphne van Leeuwen
d.van.leeuwen@cwi.nl

October 17, 2018

Executive Summary

Community detection finds its applications in many fields of research. Also in the movement of crowds, community detection can be helpful to get insight in movement patterns. For example, the city of Amsterdam has interest in getting insight in how people move around in the city, especially after opening a new metro line. However, there is no single method available that allows to find clusters best. Various metrics and algorithms exist on community detection, all having its benefits and drawbacks.

Often modularity is used as metric to find the optimal set of clusters within a set of nodes. However, when looking at a graph as a Markov Chain, where going from one node to another represents going from one state to another, the so called Kemeny constant can be used as metric as well. Intuitively, the Kemeny constant can then be interpreted as the average number of steps required going from a randomly chosen state to a another random state. Minimizing this value would then represent an effective partitioning of a graph, since the number of required steps to move around is minimized. In this study it is examined whether the Kemeny constant can be used to effectively detect clusters in a graph with every node starting as a single cluster. Additionally, various variants of quantifying the quality of the detected clusters using the Kemeny constant are compared in order to get insight in the implications of averaging, normalizing and using the uniform Kemeny constant.

This study is performed by defining six different Kemeny Constant Quality Functions (KCQF) and using three datasets. For the first dataset, which is rather small, it was possible to perform an exhaustive search. Then, all results were compared to the optimal solution. For the two other datasets the Louvain algorithm is used to find the optimal partitioning in communities. This Algorithm optimizes based on modularity. Hence, is was examined whether the KCQF metrics would behave in a similar manner as the modularity metric during the algorithm.

This study has shown that the Kemeny constant seems suitable to create clusters from single nodes. It is also shown that the way the Kemeny values of the different clusters are combined into a single quality function does make a difference in the performance. With this knowledge, further research can be performed to develop an algorithm which takes Kemeny values as criteria to form clusters. This method then can be applied to all kinds of problems, for example on the Amsterdam city data in order to get insight in the movements of crowds.

Contents

1 Introduction **3**

2 Literature review **4**

3 Data Used **6**
3.1 Description of Data 6
3.2 Processing of Data 6

4 Model Description **6**
4.1 Modularity 6
4.2 Determining Clusters using Modularity 7
4.3 Network as a Markov Chain 8
4.4 Mean First Passage Time 8
4.5 Kemeny Constant as a metric for clustering 10
4.6 Validating Clustering Results 11
4.6.1 Exhaustive search 11
4.6.2 Kemeny in Louvain Algorithm 11

5 Results **11**
5.1 Exhaustive search 11
5.2 Louvain Algorithm 12

6 Discussion **15**

7 Conclusion **17**

1 Introduction

Community detection is concerned with detection of clusters within networks and is originated from the mathematical sciences. Nowadays, it finds its applications in various fields such as computer, social and biological sciences [1]. Also in the field of crowd movement, interest in clustering algorithms on GPS data is shown [2]. Numerous methods are available for community detection, all having advantages for different types of networks [3]. In the field of clustering crowd movement, various methods and metrics are used to optimize the identification of clusters [4].

Measuring clusters in movement of people has numerous applications. One of them is identifying the movement of groups of people in order to get more insight in the behavior of crowds [5]. Cities such as Amsterdam face an increase in crowdedness and therefore want insight in how crowds move around the city [6]. Additionally, Amsterdam recently opened a new metro-line. Therefore, it would be interesting to get insight in how this new line impacts the movement of crowds in the city.

One way of determining clusters in a network is by looking at its connectivity. The idea that nodes relatively close to each other form a cluster is called *homophily* in social sciences. Intuitively, one can interpret this following a *Random Walk* process. A random walk process starts randomly at a node and proceeds randomly to other nodes following the edges in the network. Weighted edges represent the probability taking a certain edge as opposed to another. On average, when the number of steps between nodes is relatively small this means the nodes are *close* to each other. The matrix representing the weights of the edges is called the transition matrix. Then, one can see the network as a Markov Chain representing the probabilities going from one node (state) to another.

From this transition matrix, the *Mean First Passage (MFP)* times can be extracted. These MFP times represent the expected number of transitions required going from one node to another. This can be interpreted as the distance between nodes. Taking a weighted average of these MFP times results in the *Kemeny constant*. This Kemeny constant is defined as the expected number of transitions required going from a random node to any other node in a network. Hence, the Kemeny constant represents the connectivity of that network.

This Kemeny constant can be used as a metric to find clusters within a network. It is already shown that the Kemeny constant can be used to determine optimal clusters by decomposing a network into different clusters. However, determining clusters in a network using the Kemeny constant starting with every node as a single cluster requires a metric which takes multiple Kemeny values into account. This leads to the following research questions.

1. **Can the Kemeny constant be used as a metric to determine the**

optimal clusters within a network starting from a network with every node as a single cluster?

2. **What variant of combining multiple Kemeny constant values leads to clusters which are most consistent when using modularity as metric?**

This research paper is structured as follows. First an overview of relevant literature is discussed in section 2. Then, an overview of the dataset used and processing of the data is given in section 3. This is followed by a detailed description of the models and algorithm in secton 4 and the results in section 5. This research paper is concluded with a discussion and insights for future studies in section 6 and a final conclusion with the most important findings in section 7.

2 Literature review

In the scientific literature there is no consensus on what clustering method should be used for different types of networks. Additionally, various metrics are used as a criterion for performance of a clustering algorithm, such as *modularity*, *conductance*, *coverage* and many others. The performance of these measures differs based on the type of network, size of network and possibly other variables [7]. Therefore, for the field of community detection it is interesting to get insight in the strengths and weaknesses of different measures for different types of networks.

Fortunato wrote an extensive overview on what methods are available for community detection [8]. This overview also contains various extensions and limitations of the modularity metric, which is one of the most commonly used metrics. Even though this metric is proven to be used successfully[9], it also has its limitations. One of the drawbacks of the modularity metric is the so called *resolution limit*, which means modularity-based clustering algorithms fail to detect small clusters in a substantially large network [10]. Another drawback is that different partitions could have (almost) similar modularity values which makes it hard to find the global optimum. Lastly, the maximum modularity value depends on the size of the network and the number of clusters. These limitations should be taken into account when using the modularity value in practice [11]. Adaptations to this modularity optimization metric have been applied to networks where multiple edges and self-connections are not included, in order to make it more appropriate [12]. However, for other types of networks there is still room for improvement for this metric.

Earlier studies successfully used modularity as metric to optimize clusters. In a study it was examined whether clusters in Belgium appear based on telephone data. In order to detect these clusters, they developed an algorithm which efficiently maximizes the modularity value, which is called the L*ouvain Algorithm*

[13]. In Great Britain a similar study was performed, where the *spectral optimization algorithm* [14] was used to optimize the modularity value [15]. Both algorithms use modularity as a connectivity measure in order to optimize the detection of community structure.

But there are also numerous methods which use other metrics than modularity to determine communities. One way is to look at distances between nodes and find communities with the smallest total distances. A well known algorithm is to use the Random Walk Approach [16] to determine distance between nodes [17]. The idea behind this approach is that when you randomly start at a node, you spend relatively more time at nodes which belong to the community of this cluster compared to nodes which do not. Many variants of this algorithm are developed, for example the *Markov Cluster Algorithm (MCL)* [18]. This algorithm simulates the flow in a graph, with a matrix consisting of probabilities of that node going to another node as basis. After some iterations this algorithm disconnects and further connects nodes, which results in several separated communities. In this algorithm the probability of going from one node to another is used as a distance metric to get insight in the connectivity of a network.

An alternative measure for the connectivity of a network is the *Kemeny Constant*. This value is a weighted average of mean first passage times of a transition matrix, which means it represents the expected number of steps needed to reach its desired location [19]. Since we assume that taking less steps represents a better connected network, this Kemeny constant can be used as a measure for the connectivity of a network. Previous studies showed for example that small values of the Kemeny constant represented small traveling times in road networks [20] and minimizing the Kemeny constant has been used to detect anomalitites in networks the the field of Robotic Surveillance [21].

Various methods were examined to minimize the Kemeny constant value. One method is to iteratively remove the edge with the biggest impact on the Kemeny constant. This impact is calculated by taking the derivative of the Kemeny constant for that node. This so-called *Kemeny Decomposition Algorithm* [22] is tested on various benchmark graphs and showed good results on both the *Courtois matrix* [23] and the *Social Network in a Karate Club* [24] examples.

Former research also showed that it is possible to detect spatially connected clusters, even though no spatial characteristics were explicitly taken into account in the algorithm [25]. For this study, modularity was used as optimization metric. However, even though the clusters visually seem correct and robust, the modularity value was very low. Hence no objective and quantitative support was given for the communities discovered.

In this study it is examined whether communities can be identified using the Kemeny constant as metric. Additionally, this measure is optimized to generate most consistent results with respect to modularity. For this we use several

benchmark datasets to validate the performance, as well as Origin-Destination data of Amsterdam in order to conclude whether spatially connected clusters can be supported by an objective measure.

3 Data Used

3.1 Description of Data

For this study various data sets are used. In order to validate whether the Kemeny Constant can be used to identify communities, datasets with known communities are used. First, we use the *Courtois matrix* [23] since this allows to calculate the Kemeny values for all possible permutations. Hence, it can easily be determined whether the Kemeny constant allows to identify the best solution. Secondly, the *Social Network in a Karate Club* data, also known as the *Zachary matrix* [24], is used. This matrix with a known solution allows to examine how well communities are identified by heuristic methods. For example, the performance of the Louvain algorithm can be quantified by various metrics based on the Kemeny constant.

Lastly, a dataset consisting of origin-destination (OD) pairs of the Amsterdam Metropolitan region is used [26]. This empirical dataset is provided by Google, which collected these data by keeping track of movements through Android phones. The data is aggregated on both location and time. There are 512 regions defined and all data is grouped per hour and normalized. This results in a dataset with intensities between region, with values between 0 and 1. Also note that the data represents a fully connected graph.

3.2 Processing of Data

An extensive data-analysis is performed on this OD dataset [25] which revealed that there are time dependencies in the dataset. For example, the time of the day is clearly visible in the data. Moreover, day of the week, what month and other characteristics were found. However, aggregating the data resulted in robust clusters. Hence, for simplicity this aggregated dataset is used for this research as well.

4 Model Description

4.1 Modularity

One of the most commonly used measures to determine the connectivity of a network is modularity [27]. Modularity represents the fraction of edges within a community compared to the total amount of edges. The value of modularities lies between -1 and 1, where a value of 1 means all edges are within clusters, and there are no edges between clusters. A negative value means there are less

edges within clusters then one would expect on a random graph with similar properties. The definition of modularities is

$$Q = \frac{1}{2m} \sum_{vw} \left[\left(A_{vw} - \frac{k_v k_w}{2m} \right) \delta(c_v, c_w) \right] \tag{1}$$

where m is the sum of all weights in the graph, A_{vw} is the weight from node v to node w, k_v is the sum of weights of node v, k_w is the sum of weights of node w and $\delta(c_v, c_w)$ represents a function which returns 1 when node v is in the same cluster as node w, 0 otherwise. The part $A_{vw} - \frac{k_v k_w}{2m}$ represents the difference between the actual weight of an edge and the weight the edge would have on a random graph with similar properties. When considering a directed graph the definition of modularity slightly changes [28], and is defined by

$$Q = \frac{1}{m} \sum_{vw} \left[\left(A_{vw} - \frac{k_v^{in} k_w^{out}}{m} \right) \delta(c_v, c_w) \right] \tag{2}$$

where k_v^{in} represents the sum of weights of the edges going to v and k_w^{out} represents the sum of weights going from w.

4.2 Determining Clusters using Modularity

One could use modularity to identify clusters in a network. For small networks this can be done by calculating the modularity of all possibilities and maximizing this values. However, for many networks this is an impractical approach. Therefore, a heuristic method is needed to optimize the modularity value. An efficient heuristic method is the *Louvain ALgorithm* [29] which is easily extended to allow for directionality [30]. The Louvain algorithm works as follows and is visualized in Figure 1

All nodes start as a single community. Then it iteratively performs the following two steps.

1. It runs randomly over all nodes and the modularity gain in calculated when that node is added to a neighboring community. The node is added to the neighboring community with the largest increase in modularity. This steps runs sequentially over all nodes until no more modularity increase can occur. This step is visualized going from state **A** to state **B** in Figure 1.

2. All nodes belonging to the same community are combined into a single node. All weights of the nodes are summed which means the weights of the nodes within the community become the weight of the new node to itself. The weights to and from nodes in other communities are summed as well and become the weights of the edges between the new nodes. This step is visualized going from state **B** to state **C** in Figure 1.

After step 2, both steps are repeated until the outcome is the same as the iteration before. Hence, either the number of communities reduces further (is this leads to a higher modularity value) or the same communities remain which leads to the termination of the algorithm.

4.3 Network as a Markov Chain

Another view on a network is to see it as a Markov Chain. When normalizing the transition matrix, the probabilities going from one node to another appear. This can be interpreted as follows. Assuming that you will always move from a node (note that a move can also be a self loop), the probability of moving is 1. When you calculate the fraction of the weight to each node of the sum of weights of the nodes it can go to, this can be interpreted as the probability going to that node. With these probabilities, one can define a Markov Chain P on node set $\mathbb{S}$. The weighted directed graph is then called the *Markov influence graph*. The *Deviation Matrix* is then given by

$$D_p = (I - P + \Pi_p)^{-1} - \Pi_p \tag{3}$$

where I is the identity matrix and $\Pi_P = (\Pi_P(i,j))_{\mathbb{S}x\mathbb{S}}$ is called the *ergodic projector* and is given by

$$\Pi_P = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} P^n \tag{4}$$

in which $\Pi_P(i,j)$ gives the long-term average number of visits of the Markov chain from i to j [22].

4.4 Mean First Passage Time

One interesting aspect that can be derived from the deviation matrix is the *Mean First Passage times (MFP)*. The MFP is the mean number of transitions required to go from one node to another. Therefore, it is a way to represent the distance between nodes. The Mean First Passage matrix is defined by

$$M = (I - D_p + \bar{1}\bar{1}^T \cdot dg(D_p)) \cdot dg(\Pi_P)^{-1} \tag{5}$$

where $\bar{1}$ is an appropriately sized vector and dg() means taking the matrix where the non-diagonal values are set to zero. With this matrix, one can calculate the *Kemeny Constant*, which represents the weighted average of the mean first passage times. The Kemeny Constant is defined by

$$K_p = \sum_{j\in\mathbb{S}} M(i,j)\pi_P(j), \forall i \in \mathbb{S} \tag{6}$$

where π_P is the unique stationary distribution of P. As can be noticed from equation 6, this value is independent of the starting position i. In other words,

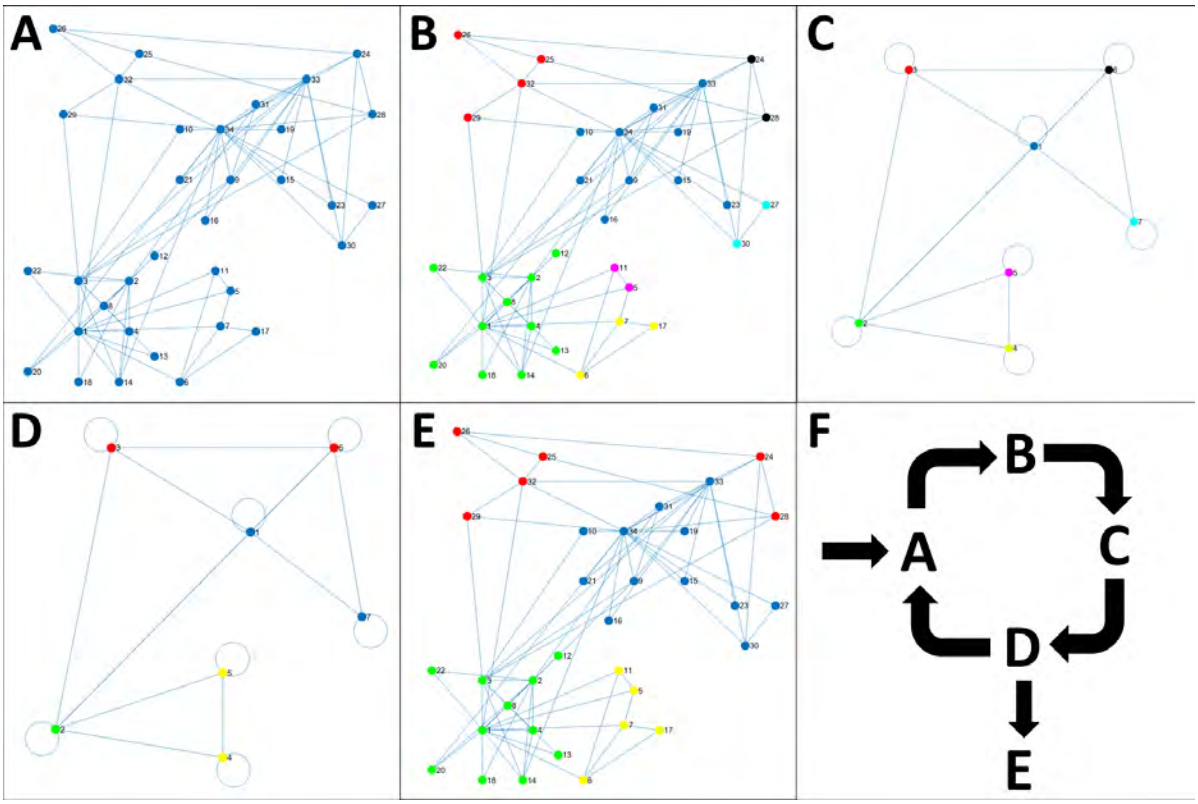

Figure 1: Illustration of the Louvain Algorithm on the Zachary Matrix. **A.** The matrix visualized as a graph with which is used as input for the algorithm. **B.** Seven clusters are formed after step one of the algorithm. **C.** The seven clusters have become seven separate nodes after finished step 2 and become input for step 1 of the second iteration of the algorithm. **D.** After step 1 of the second iteration, new clusters are formed. **E.** After merging the clusters into new nodes, no increase in modularity occured. Hence, the clusters found in **D** represent the best solution. This clustering applied on the original matrix results in these four clusters. **F.** Schematic view on the algorithm.

the expected number of steps before reaching a randomly chosen node is constant and independent of the starting node. To give some intuition, the value of the Kemeny constant represents the average number of steps needed going from one node to another. As a measure of the connectivity, small values of K_P would indicate a better connectivity compared to a bigger value of K_P.

4.5 Kemeny Constant as a metric for clustering

This property of the Kemeny constant allows to quantify the quality of identified communities. However, when there is more than one cluster identified, multiple values of K_P exist and in order to quantify the quality of all clusters identified, the different values need to be combined. Numerous metrics are possible in order to quantify the quality of the communities. In this study 6 different *Kemeny based Clustering Quality Functions (KCQF)* are tested, based on two main approaches: taking a normalized mean and taking into account the uniform Kemeny constant.

Intuitively the mean of all Kemeny values would be suitable (equation 9). But one might also need to take into account the size of the clusters, since bigger clusters tend to have larger Kemeny values. Hence, normilization is applied in two different manners (equation 7 and 8). The other main approach is to take into account the uniform Kemeny constant. This value represent the average number of steps required when the connections within the cluster are randomly assigned. Hence, the difference between the Kemeny value and the uniform Kemeny value represents the connectivity of the cluster (equation 11). Also for this metric a normalization is applied (equation 10). Lastly, the fraction between the Kemeny value and the uniform Kemeny value is used as metric (equation 12).

All metrics can be formalized as follows.

$$KCQF\_1 = \frac{1}{N} \sum_{j \in C} \frac{K_j}{S_j^2} \tag{7}$$

$$KCQF\_2 = \frac{1}{N} \sum_{j \in C} \frac{K_j}{S_j} \tag{8}$$

$$KCQF\_3 = \frac{1}{N} \sum_{j \in C} K_j \tag{9}$$

$$KCQF\_4 = \frac{1}{N} \sum_{j \in C} \frac{K_j - Kunif_j}{S_j^2} \tag{10}$$

$$KCQF\_5 = \frac{1}{N} \sum_{j \in C} K_j - Kunif_j \tag{11}$$

$$KCQF\_6 = \frac{1}{N} \sum_{j \in C} \frac{K_j}{Kunif_j} \tag{12}$$

where N is the number of clusters, C represents the clusters identified, K_j is the Kemeny value of each cluster and S_j is the size of each cluster. $Kunif$ is the Kemeny value calculated from the uniform matrix of each cluster.

4.6 Validating Clustering Results

4.6.1 Exhaustive search

For small matrices, the Kemeny value of all possibilities can be calculated. It is expected that the solution with the smallest Kemeny value should be the optimal solution. For the Courtois matrix the solution is known, which allows to validate whether the various Kemeny based Clustering Quality Functions lead to the optimal solution.

4.6.2 Kemeny in Louvain Algorithm

For bigger matrices, the Louvain algorithm is used to identify clusters. The Louvain algorithm works based on the maximization of the modularity value. So, one would expect that the Kemeny value would decrease accordingly. Hence, every iteration the various values of KCQF are calculated and expected to decrease.

In addition, during step 1 of the algorithm, the value of modularity is forced to either increase or remain the same (this is by definition of the algorithm). Hence, one would expect the Kemeny value to decrease or remain the same accordingly. To examine this, the development of the Kemeny value during step 1 of the algorithm is visualized in order to get insight in the performance of the Kemeny value as measure for the clustering quality.

5 Results

5.1 Exhaustive search

For the Courtois matrix it can be seen that only $KCQF\_6$ results in the optimal solution (see Table 1). It also shows that based on modularity not the optimal solution is reached. $KCQF\_2$ and $KCQF\_3$ result in a solution somewhat similar to the optimal solution and equal to the modularity solution. $KCQF\_4$ is almost similar to the modularity solution, except from the last cluster. $KCQF\_1$ and $KCQF\_5$ show solutions with only two clusters. $KCQF\_5$ and $KCQF\_1$ reflect the first two clusters of the solution. However, the third cluster of the solution is classified as either one of the first two cluster. In Table 2 the values for the different metrics for various amounts of clusters are shown, as well as the optimal value on which the final solution is based.

Table 1: Community structure based on exhaustive search on the Courtois matrix optimized for Modularity (Mod) and six different KCQF values.

Solution	**Mod**	**KCQF_1**	**KCQF_2**	**KCQF_3**	**KCQF_4**	**KCQF_5**	**KCQF_6**
1	1	1	1	1	1	1	1
1	2	1	2	2	2	1	1
1	2	1	2	2	2	1	1
2	3	2	3	3	3	2	2
2	3	2	3	3	3	2	2
3	4	2	4	4	4	1	3
3	4	1	4	4	4	2	3
3	4	1	4	4	1	1	3

Table 2: The values of Modularity (Mod) and KCQF for various numbers of clusters. Highlighted are the values which are either maximized (Modularity) or minimized (KCQF) which correspond to the solutions in Table 1.

Nr clusters	**Mod**	**KCQF_1**	**KCQF_2**	**KCQF_3**	**KCQF_4**	**KCQF_5**	**KCQF_6**
1	0	64,15	513,17	4105,35	0	0	1
2	0,47	0,59	2,37	9,49	-113,30	-1752,06	0,01
3	0,66	0,60	1,59	4,40	-135,44	-974,95	0
4	0,68	0,71	1,27	2,64	-146,28	-585,39	0,01

5.2 Louvain Algorithm

For the larger matrices the Louvain algorithm is used to test whether the Kemeny value decreases along with the increase of modularity. The results for the Zachary matrix are shown in Table 3. It can be seen that only $KCQF\_1$, $KCQF\_2$, $KCQF\_5$ and $KCQF\_6$ show a decrease in the Kemeny value.

Table 3: Iterations of the Louvain algorithm on the Zachary matrix. The algorithm optimizes based on the modularity increase. Also the change in Kemeny values is shown for the various KCQF values. Highlighted are the metrics which decrease compared to the previous iteration.

Iteration	**Mod**	**KCQF_1**	**KCQF_2**	**KCQF_3**	**KCQF_4**	**KCQF_5**	**KCQF_6**
1	0,35	0,74	2,09	7,78	-0.47	-6,18	0,58
2	0,44	0,19	1,41	11,85	-0,24	-12,27	0,48
3	0,44	0,19	1,41	11,85	-0,24	-12,27	0,48

When looking into more detail it becomes clear how the values of the different metrics develop in the first step of the Louvain algorithm. By definition, the value of modularity should either increase or remain the same. In Figure 2 the development of the Kemeny metrics are shown. It can be seen that all

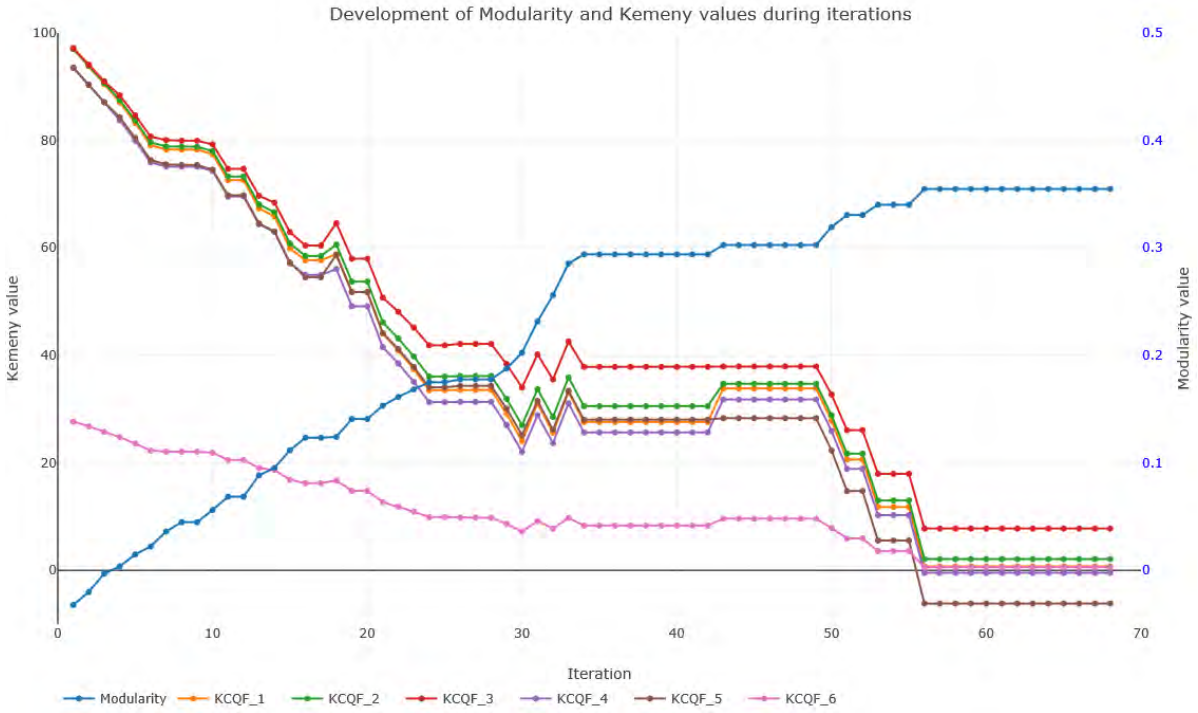

Figure 2: The development of the Modularity and KCQF metrics during the first iteration of the Louvain algorithm on the Zachary data. All KCQF metrics correspond to the y-axis on the left (Kemeny value). The modularity value corresponds to the y-axis on the right. On the x-xis the different steps in the first iteration are shown.

Kemeny values show qualitatively similar behavior, meaning that the lines either increase, decrease or remain the same in a similar fashion. However, when looking closely to the iteration steps between iteration 40 and 50, one can see that $KCQF\_3$ and $KCQF\_5$ remain the same whereas all other metrics show a slight increase. In that sense $KCQF\_3$ and $KCQF\_5$ would follow the modularity metric better than the other ones. But in general, for all Kemeny metrics the values decrease when modularity increases.

For the Amsterdam data, the results are shown in Table 4. For this data only $KCQF\_1$ and $KCQF\_2$ show a decrease in Kemeny value. When looking into more detail, $KCQF\_1$, $KCQF\_2$, $KCQF\_4$, $KCQF\_5$ (see Figure 3) and $KCQF\_6$ (see Figure 4) show similar behavior as expected. However, $KCQF\_3$ shows increasing behavior which indicates the lack of normalizing for the size of each cluster causes the increase of the Kemeny metric. In other words, for this dataset it is necessary to normalize for clustersize. Apparently the decrease of the sum of Kemeny constants of all clusters is less than $\frac{1}{N}$ of the previous sum.

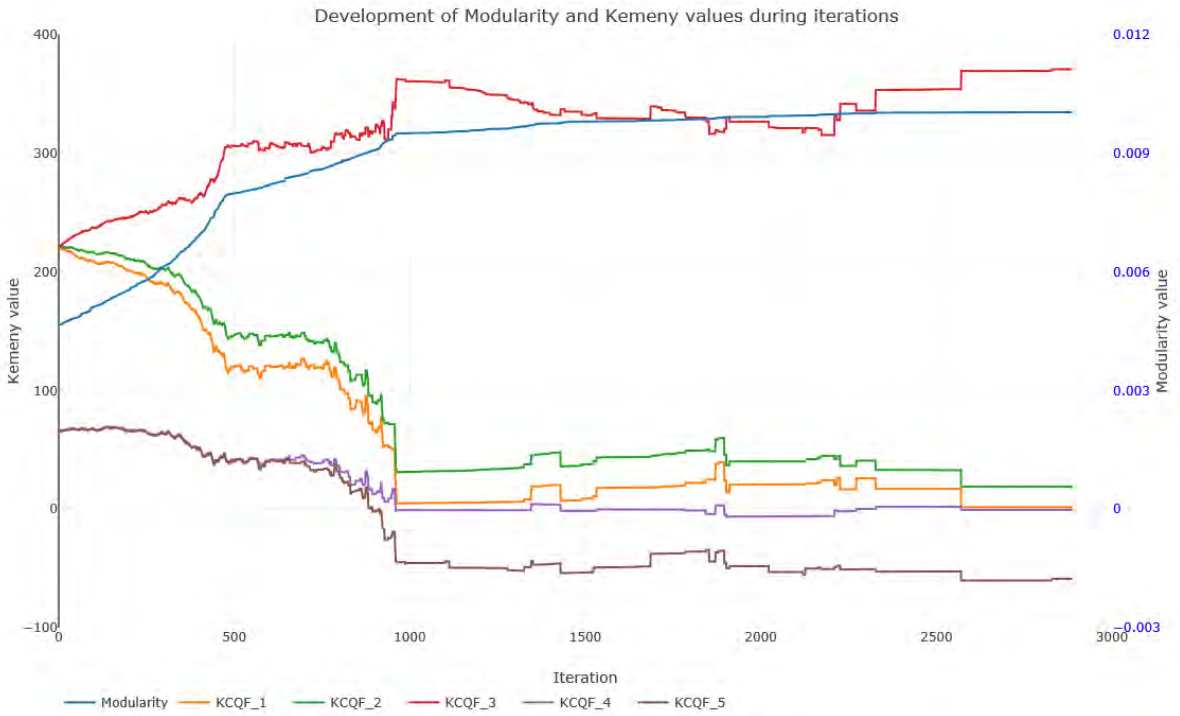

Figure 3: The development of the Modularity and the first five KCQF metrics during the first iteration of the Louvain algorithm on the Amsterdam data. All KCQF metrics correspond to the y-axis on the left (Kemeny value). The modularity value corresponds to the y-axis on the right. On the x-xis the different steps in the first iteration are shown. $KCQF\_6$ is not shown due to a different scale on the y-axis which impaired visibility.

Table 4: Iterations of the Louvain algorithm on the Amsterdam data. The algorithm optimizes based on the modularity increase. Also the change in Kemeny values is shown for the various KCQF values. Highlighted are the metrics which decrease compared to the previous iteration.

Iteration	**Mod**	**KCQF_1**	**KCQF_2**	**KCQF_3**	**KCQF_4**	**KCQF_5**	**KCQF_6**
1	0,0100	1,6545	18,8945	370,9738	-0,6443	-58,6830	0,8551
2	0,0104	1,2452	15,5278	391,8439	-0,2968	-50,3522	0,8810
3	0,0104	1,2452	15,5278	391,8439	-0,2968	-50,3522	0,8810

When combining the results from both datasets, $KCQF\_1$ and $KCQF\_2$ show most promising results. Both metrics show a decrease during the Louvain algorithm. Also in the detailed view of the first iteration step both metrics show a decrease in general. However, it should be noted that in the detailed view

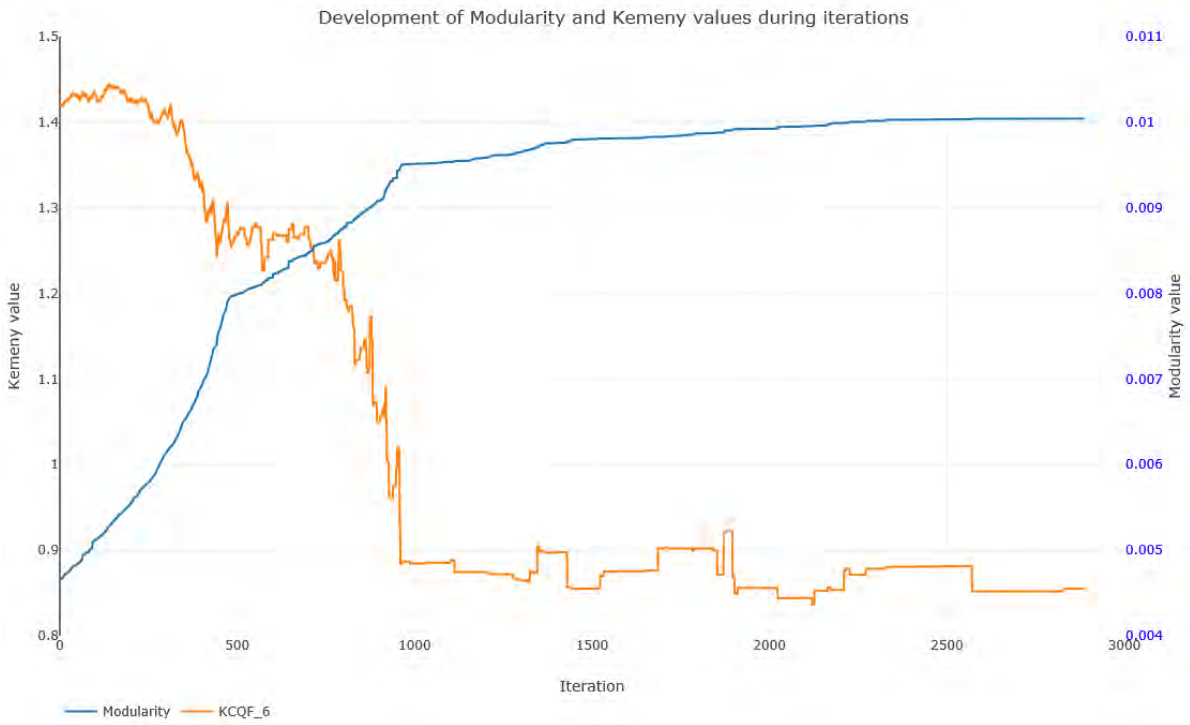

Figure 4: The development of the Modularity and $KCQF\_6$ metric during the first iteration of the Louvain algorithm on the Amsterdam data. The $KCQF\_6$ metric corresponds to the y-axis on the left (Kemeny value). The modularity value corresponds to the y-axis on the right. On the x-xis the different steps in the first iteration are shown.

these metrics perform worse than $KCQF\_5$. $KCQF\_3$ and $KCQF\_4$ show an increase in both cases where a decrease is expected and are therefore the worst metrics. $KCQF\_5$ and $KCQF\_6$ only show a decrease for the Zachary data.

6 Discussion

Detecting communities in a dataset when starting with all nodes as single clusters is often done using the modularity value. However, using the properties of the Kemeny constant seems also promising to detect clusters. But when doing so, one needs to come up with a metric which combines the Kemeny values of the different clusters into one single metric which reflects the quality of the detected communities. In this study, two types of metrics were tested: the average of Kemeny values of all clusters and using the uniform Kemeny values in order to relate the clusters found to random clusters of similar size. For both types, different manners of normalizing were applied and compared in order to

test what metrics are useful and to gain insight in when to use what metric as quality function to detect communities.

When combining the results from the three different datasets, the quality metric where the average is normalized by dividing it by the size of the cluster performs most consistently with respect to the modularity. Comparable metrics, where is normalized with the squared cluster size or without normalization perform worse. When there is no normalization, the decrease of the Kemeny value does not compensate for the decrease in the total amount of clusters which causes the total to increase instead of decrease. On the other hand, when you emphasize the number of nodes within a cluster, by normalizing with the square of cluster size, the number of clusters becomes too small. This is because a small number of clusters results in larger clusters, as can be seen in the exhaustive search result in section 5.1.

Concerning the metrics including the uniform Kemeny value, the way of normalizing seems to be of great importance. As with the normalized average metrics, when the difference between the Kemeny value and the uniform Kemeny value is normalized by the squared cluster size, this cluster size makes too much impact. In this case the difference between the Kemeny value and the uniform Kemeny value is always negative. Dividing by the squared cluster size causes larger negative values when cluster sizes are smaller, since the number of clusters is bigger. On the other hand, no normalization results in a too small amount of clusters. Dividing the Kemeny value by the uniform Kemeny value results in the optimal solution in the exhaustive search and also on the Zachary data it shows good results. However, on the Amsterdam data it shows no consistency with the modularity, just as the other metrics with the uniform Kemeny value.

It is hard to conclude which of these metrics performs best, since the performance is different on the various datasets. Additionally, performance on the exhaustive search is different from the Louvain algorithm. And lastly, some assumptions and practical choices were made which may have influenced the result. For example, The Zachary data does not contain any self-connection, whereas the Amsterdam data does. When the Louvain algorithm is initialized for the Zachary data, all nodes are separate clusters with a connection value of 0. For these clusters it is not possible to calculate a Kemeny value. For this study this value is arbitrarily set to 100. If a generic quality measure is developed, one could make this value dependent on for example the number of clusters to make it scalable for larger matrices.

Another assumption that is made is for the calculation of the uniform matrix. The way the uniform matrix is calculated results in a matrix with self-connections. When the Louvain algorithm is initialized, it is possible to calculate the uniform Kemeny values, whereas the Kemeny values for the same data become the arbitrarily chosen 100. One should take this into account when interpreting the outcome. Another possibility to correct for this is to add self

connection to the input data, for example by applying the random surfer [31].

7 Conclusion

Community detection finds its applications in many fields of research. Most often, modularity is used as metric to find the optimal set of clusters within a set of nodes. When looking at a graph as a Markov Chain, where going from one node to another represents going from one state to another, the Kemeny constant can be used as metric as well. The Kemeny constant can then intuitively be interpreted as the average number of steps required going from a randomly chosen state to a another random state. A low value would then represent an effective partitioning of a graph, since the number of required steps to move around is minimized. This study has shown that the Kemeny constant seems suitable to create clusters from single nodes. It is also shown that the way the Kemeny values of the different clusters are combined does make a difference in the performance of the community detection. With this knowledge further research can be performed to develop a Louvain-like algorithm which takes Kemeny values as criteria to form clusters. In conclusion, the Kemeny constant is promising to use as metric to detect communities within graphs. When one takes into account the implications of the various manners to combine the Kemeny values into a quality function, the Kemeny constant is suitable for community detection.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, 2002.

[2] J. Chen, T. Hu, P. Zhang, and W. Shi, "Trajectory clustering for people's movement pattern based on Crowd souring data," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2014.

[3] S. Fortunato and D. Hric, "Community detection in networks: A user guide," 2016.

[4] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering Personal Gazetteers: An Interactive Clustering Approach."

[5] D. Helbing, G. Tröster, M. Wirz, and D. Roggen, "Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods," *Networks and Heterogeneous Media*, 2011.

[6] "Stadsenquête Drukte en Balans 2017, Samenvatting," Onderzoek, Informatie en Statistiek, Tech. Rep., 2017.

[7] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki, "Is there a best quality metric for graph clusters?" in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011.

[8] S. Fortunato, "Community detection in graphs," 2010.

[9] J. Ruan, "A Fully Automated Method for Discovering Community Structures in High Dimensional Data."

[10] S. Fortunato, M. Barthélemy, and M. Barth, "Resolution Limit in Community Detection," *Source: Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007. [Online]. Available: http://www.jstor.org/stable/25426046http://about.jstor.org/terms

[11] B. H. Good, Y.-A. De Montjoye, and A. Clauset, "The performance of modularity maximization in practical contexts."

[12] C. P. Massen and J. P. K. Doye, "Identifying communities within energy landscapes," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 2005.

[13] V. Blondel, G. Krings, I. T. Traducteur, J. Corrigan, and I. Thomas, "Brussels Studies Regions and borders of mobile telephony in Belgium and in the Brussels metropolitan zone." [Online]. Available: http://brussels.revues.org/806http://

[14] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, 2006.

[15] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton, and S. H. Strogatz, "Redrawing the map of Great Britain from a network of human interactions," *PLoS ONE*, 2010.

[16] B. Hughes, *Random walks and random environments*, 1995.

[17] H. Zhou, "Distance, dissimilarity index, and network community structure," 2003.

[18] S. van Dongen, "GRAPH CLUSTERING," Ph.D. dissertation, Dutch National Research Institute for Mathematics and Computer Science, University of Utrecht, Netherlands, 2000.

[19] M. Catral, S. J. Kirkland, M. Neumann, and N. S. Sze, "The Kemeny constant for finite homogeneous ergodic Markov chains," *Journal of Scientific Computing*, 2010.

[20] E. Crisostomi, S. Kirkland, and R. Shorten, "A Google-like Model of Road Network Dynamics and its Application to Regulation and Control," 2010.

[21] P. Agharkar, R. Patel, and F. Bullo, "Robotic surveillance and Markov chains with minimal first passage time," in *Proceedings of the IEEE Conference on Decision and Control*, 2014.

[22] J. Berkhout and B. F. Heidergott, "Analysis of Markov Influence Graphs," 2018.

[23] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[24] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *Journal of Anthropological Research*, 1977.

[25] D. Van Leeuwen, J. Bosman, and E. Dugundji, "ScienceDirect Spatio-Temporal Clustering of Time-Dependent Origin-Destination Electronic Trace Data," *Procedia Computer Science Portugal Procedia Computer Science*, vol. 130, no. 00, pp. 359–367, 2018. [Online]. Available: www.elsevier.com/locate/procedia

[26] D. Van Leeuwen, "Network partitioning on Origin-Destination traces."

[27] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," 2003.

[28] E. A. Leicht and M. E. J. Newman, "Community structure in directed networks," 2007.

[29] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[30] N. Dugué, A. Perez, and N. Dugué Anthony Perez, "Directed Louvain : maximizing modularity in directed networks," 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01231784

[31] M. Levene and G. Loizou, "Kemeny's constant and the random surfer," *American Mathematical Monthly*, 2002.