A High Resolution Clique-based Overlapping Community Detection Algorithm for Small-world Networks

András Bóta
University of Szeged, Institute of Informatics Address,
P. O. Box 652., 6701 Szeged, Hungary
E-mail: bandras@inf.u-szeged.hu

Miklós Krész
University of Szeged, Juhász Gyula Faculty of Education,
Boldogasszony bvd. 6, 6720 Szeged, Hungary
E-mail: kresz@jgypk.u-szeged.hu

Keywords: network science, community detection, overlapping communities

Received: June 24, 2013

In this paper we propose a clique-based high-resolution overlapping community detection algorithm. The hub percolation method is able to find a large number of highly overlapping communities. Using different hub-selection strategies and parametrization we are able to fine tune the resolution of the algorithm. We also propose a weighted hub-selection strategy, allowing the algorithm to handle weighted networks in a natural way, without additional filtering. We will evaluate our method on various benchmarks, and we will also demonstrate the usefulness of our algorithm on a real-life economic case-study.

Povzetek: Predstavljena je nova hevristika za reševanje evklidskega BDMST problema. Primerjalni testi pokažejo prednosti pred obstoječimi metodami.

1 Introduction

One of the landmarks in graph theory was the introduction of small-world networks by Watts and Strogatz [31]. They have observed, that in real-life networks, the typical distance between two randomly chosen nodes grows proportionally to the logarithm of the number of nodes in the network. Since then, several other properties of real-life networks was discovered. The degree distribution of these networks follows a power-law [2], and the edge distribution is not only globally, but also locally inhomogeneous. This latter feature is called community structure [11]. The goal of community detection is the discovery of this structure. While the phenomenon of communities is well observed, an exact definition is difficult to find.

In recent years, a large number of community detection algorithms have been proposed. Most of these consider communities to be disjoint vertex sets, and adopt the following intuition: They are looking for a partitioning of the nodes, which maximizes the number of edges between the nodes inside the sets, and minimizes them between the sets. It is also a goal to find meaningful communities, i.e. they discard trivial solutions of the problem (like a single community containing all of the vertices). Newman proposed modularity [27] as an efficient way to measure the goodness of disjoint communities. A comprehensive review of community detection can be found in [10].

The traditional definition of community allows disjoint vertex sets only. Based on the observation that in real-life networks, nodes can belong to multiple communities, Palla et al. introduced the concept of overlapping community detection and proposed the clique percolation method [29] as a solution. The idea of finding maximal cliques and joining them according to some criteria is the basis of several overlapping community detection algorithms [17, 21]. Other approaches are based on block models [12, 7], edge clustering [9, 1], label propagation [13] or optimization according to some fitness function [25, 20].

Measuring the goodness of overlapping community detection algorithms is complicated, since there is no agreement on the definition of an overlapping community. The specifications of different applications depend mainly on the ratio of overlaps between communities: several approaches require only a loose relaxation of the original ”non-overlapping” definition in such way that occurence of nodes belonging to multiple communities is strongly restricted [13]; other concepts prefer highly overlapping community structure [20]. The resolution of the methods are closely tied to the ratio of overlaps. A highly overlapping community structure is often associated with a large number of relatively small communities, however the opposite is not always true. Hierarchical or multiresolutional methods combine these approaches.

Corresponding to this, the output of the above mentioned algorithms can be fundamentally different. There are basically two types of evaluation in the literature: one can use some kind of benchmark network like in [18, 22, 26, 32], and compare the results to the already known community structure of the network [14, 28]. Another option is using a real-life application as an example, similar to a case-study.

In this paper, the authors propose the hub percolation overlapping community detection method. A node, that is a member of many adjacent cliques is considered more important. We refer to these nodes as hubs. We expand and join cliques if they contain the same hubs. One of the advantages of this method is, that both the hub selection and the joining criteria is adjustable. This allows us to discover different kinds of community structures from large, loosely overlapping groups to ones with a dense, highly overlapping structure. We also propose a hub-selection strategy able to handle weighted networks in a natural way without the need for filtering or pruning edges. Finally, we will rely on the framework proposed by Pluhár et al. in [3], and show how several popular algorithms can be represented in it.

We will use well-known benchmark networks [29, 26, 11] to demonstrate the difference between hub selection strategies in terms of community sizes, the size of overlaps and the number of singletons: nodes without communities. Then we will evaluate the performance of our method in two different ways. We will use the community based graph generator of Lancichinetti and Fortunato [18] to compare the results of our method to the OSLOM algorithm of the same authors [20], the COPRA method of Gregory [13], and the clique percolation method of Palla et al. [29]. We will also present a case study: we will examine the communities of an economic network constructed from the Hungarian company register. We will focus our attention on three aspects of the companies: the geographical location of them, the industrial sector they belong to and the age of the companies.

2 General framework

The authors of [3] described a general framework for overlapping community detection. In this section we summarize their approach. Here, and throughout the paper, by a graph G we mean an undirected simple graph with vertex (or node) set $V(G)$ and edge set $E(G)$. Edges might have arbitrary weight.

According to the framework introduced in [3], most community detection algorithms consist of two phases. Taking an input graph G, the first phase constructs a hypergraph $\mathcal{F} = (V, \mathcal{H})$, where $V(\mathcal{F}) = V(G)$, and $\mathcal{H} \subset 2^V$. The elements of $\mathcal{H}$ are considered the building blocks of communities. The second phase adds a distance function d to set $\mathcal{H}$, creating a metric space $\mathcal{M} = (\mathcal{H}, d)$. Using function d, a clustering algorithm creates a set of clusters $\mathcal{C}$. Finally, the arising clusters are associated to the subsets of V such that $K_i = \cup_{H \in C_i \in \mathcal{C}} H$, where K_i, the ith community corresponds to C_i, the ith cluster and K_i is just the union of the vertex set of those hyperedges that belong to C_i.

It is easy to show, that this framework applies to most community detection algorithms. In the case of the clique percolation method [29], $\mathcal{H}$ contains the k-cliques[1] of the original graph, and function d is:

$$d(K_i, K_j) = \begin{cases} 1, & \text{if } |K_i \cap K_j| = k - 1, \\ \infty, & \text{otherwise} \end{cases}$$

In the same paper [3], the authors have proposed the N^{++} community detection algorithm with a general distance function, where $\mathcal{H}$ is the same as above and $d(K_i, K_j) = 1$ only if $|K_i \cap K_j| \geq c$, where c is a parameter of the algorithm. In other cases $d(K_i, K_j) = \infty$. This method has proven its usefulness in applications [6, 16].

It is also possible to describe non-clique based methods using this formulation. In the case of COPRA [13], each element of $\mathcal{H}$ initially only contains one unique vertex $v \in V(G)$. In the second phase, these are joined according to a belonging coefficient. A threshold is introduced to provide a lower bound for community membership.

3 The hub percolation method

The motivation for creating an advanced community detection algorithm came from our previous work with the general framework of community detection [3]. Our aim was to create a flexible clique-based method taking into consideration our experiences with the clique percolation method [29] and the N^{++} method [3]. Much of the details of the algorithm described in this section comes from experiences gained during test runs on well-known benchmark networks like [32, 29, 26, 22].

The hub percolation method has two simple ideas at its core. A natural property of most approaches for overlapping community detection[2] is that cliques (fully connected subgraphs) are considered to be the purest communities. Therefore our method uses cliques at the beginning of the building process. An important observation on real-life networks is, that inside a community some members are more important than others with respect to the role of the nodes in connecting different communities. We will denote these nodes as hubs. In the building process the cliques of the graph are extended according to a limited percolation rule: two k-cliques are joined if they share $k - 1$ vertices. As a result of this process, the set of extended cliques consists of the building blocks of community detection. The joining phase of our method merges these extended cliques if they share the same hubs. Considering these ideas, an outline of the hub percolation algorithm is as follows:

1. Find the set C of all maximal cliques of size greater or equal than 3 on graph G.
2. Select the set of hubs H.
3. Create the set of extended cliques C'.

[1] Fully connected subgraphs containing exactly k nodes.

[2] In the following chapters, we will refer to overlapping community detection simply as community detection.

4. Compute the set of communities K: Take the union of extended cliques if one of them contains all the hubs in the other one.

Finding the set of all maximal cliques in a graph is a well-studied NP-hard problem of graph theory. Unfortunately an n-vertex arbitrary graph may contain i$3^{n/3}$ maximal cliques in the worst case [24]. Because of their unique structure, this number is significantly lower in small world networks allowing algorithms like in [30, 4, 8] to list the set of maximal cliques in reasonable time even for large networks. In this work we used the modified Bron-Kerbosch algorithm described in [8].

The hub selection strategy is an important part of the algorithm. Hubs represent the locally important nodes in the network. As a consequence, whether a node is a hub should depend on the t-neighborhood[3] of the given node, where t is a small number. In our interpretation hubs connect communities, therefore the deciding factor in hub selection should be the number of cliques the vertex belongs to. Each node v is assigned a hub value h_v according to the above rule, then some of them are selected if their value is higher than the average or median hub values in their t-neighborhood. It is also possible to extend the selection strategy to weighted networks. We will discuss hub selection in the next subsection.

In our method, cliques of the network are extended with a a one-step percolation rule, then merged if they share the same hubs. Introducing the filtering parameter $k \geq 2$, let us consider all cliques of size equal to k on the subgraph induced by the set of hubs H. We will denote the set of k-cliques on $G[H]$ as C_H. Then, we expand the elements of C_H according to a one-step percolation rule. Let C_e denote the set of merged cliques $c_e = c_H \cup c_0 \cup \cdots \cup c_\ell$ with $c_H \in C_H$, $c_0, \ldots, c_\ell \in C$ and $|c_0 \cap c_H| \geq 2, \ldots, |c_\ell \cap c_H| \geq 2$ [4].

The last step of our method corresponds to the joining phase of the community detection framework. We merge elementary communities if they contain the same hubs, more precisely, we take the union of two elementary communities c_{e_0} and c_{e_1} if $c_{e_0} \cap H \subseteq c_{e_1} \cap H$. We iterate this process by adding the new merged clique to C_e and removing the original ones. At the end of the process C_e contains the communities of graph G. Note, that depending on the hub selection strategy C_e may contain duplicate members, the merging process eliminates this problem as well. Each element of C_e is a union of the cliques of G and contains at least k hubs. We will refer to the members of C_e as elementary communities.

The hub percolation method
Input : Graph G, parameter k

1. Find all maximal cliques of graph G using any exact algorithm or heuristic. Let C denote the set of cliques.
2. For all $v \in V(G)$, let $h_v = |H_v|$, $H_v = \{h | v \in h, h \in C\}$.
3. Select the set of hubs H according to the hub selection strategy.
4. Let C_H denote the set of k-cliques on the subgraph induced by the hubs $G[H]$.
5. Create the set of extended cliques C_e according to the following rule: for all $c_H \in C_H$ find all cliques $c \in C$ where $|c \cap c_H| \geq 2$. Let $c_0, \ldots, c_\ell$ denote the cliques satisfying this criterion. Create the union of cliques $c_e = c_H \cup c_0 \cup \cdots \cup c_\ell$, and add c_e to C_e.
6. For all $c_{e_0}, c_{e_1} \in C_e$ add the union of them to C_e if $c_{e_0} \cap H \subseteq c_{e_1} \cap H$, and remove c_{e_0} and c_{e_1} from C_e. Iterate until there are no more merges.
7. The set C_e contains the communities of graph G.

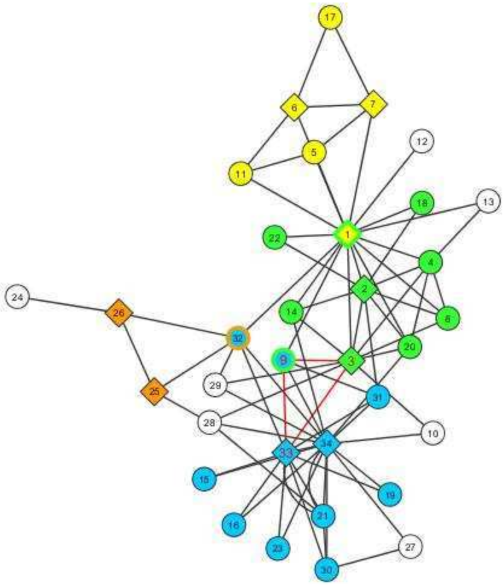

Figure 1: The communities of Zachary's karate club network [32]. Hubs are marked as diamond shapes. Nodes with multiple colors indicate overlapping nodes. The median hub selection strategy was used with $k = 2$. Nodes $9, 3, 33$ form an additional community and node 9 belongs to three communities.

It is easy to see, that the general framework proposed in applies to the hub percolation method. The edges of the hypergraph correspond to the extended communities in C_e, while the distance function is

$$d(K_i, K_j) = \begin{cases} 1, & \text{if } K_i \cap H \subseteq K_j \cap H \text{ or} \\ & K_j \cap H \subseteq K_i \cap H, \\ \infty & \text{otherwise} \end{cases}$$

[3] A t-neighbothood of a vertex v is the set of vertices N_v^t, where $u \in N_v^t$ only if the length of the shortest path from u to v is less or equal then t.

[4] Our experiances on various benchmark networks indicate that this value gives the best performance.

The community structure of Zachary's karate club network [32] can be seen on Figure 1. This network is a well-known social network, that represents friendships between the members of the club. Our method identifies five communities[5], the most interesting ones being the green and blue ones, as well as the one represented by the red triangle. During Zachary's observation the club split into two parts, because some friendships were broken. Most community detection algorithms are able to identify these subgroups even before the actual split. In our case the borders of the green and blue communities represent the borders between the two subgroups, and the red group identifies the edge that was broken when the split occurred.

3.1 Hub selection strategies

Hub selection is a crucial part of the algorithm. As we have mentioned before, each node in the graph is assigned a hub value based on the number of cliques it belongs to. Based on this value the selection strategy chooses the set of hubs H. Hubs represent "locally important" nodes so the criterion of the hub property of nodes or "hubness" should depend only on the tight neighborhood of the node. In our interpretation, this criterion depends on some simple statistical property of the first or second neighborhood of the given node, namely the average or median of neighboring hub values.

At the beginning of our work on several famous benchmark networks [26, 32] we have quickly found out, that the 2-neighborhood strategies are often not robust enough to select the appropriate hubs: hubs were relatively rare, which resulted in small overlaps and a larger than acceptable number of nodes without community memberships. A general experience was, that hubs should be "common enough", so that most of the nodes have one or more in their direct neighborhood.

Considering this, the 1-neighborhood median selection strategy provided the best community structure on these benchmark networks. This may not be the case, however, with other real-life networks. In order to extend our algorithm to handle different kinds of networks, we can generalize suggest another hub selection rule. Still considering only the direct neighborhood of nodes, we calculate the average hub value and multiply it with a parameter $q > 0$. If the hub value of the node is higher than the mean, we select it as a hub. This approach makes hub selection more flexible, allowing the algorithm to adapt to different requirements. A small value of q selects higher number of hubs resulting in larger communities with greater overlaps, while increasing q has the opposite effect. This also allows the algorithm to discover several layers of community structure on the network.

Finally, hub selection can be extended to weighted graphs in a natural way. As before, the hub value of a node is the number of cliques it belongs to. Then the values are multiplied with the strength[6] of the node. After this, the process is the same as in the previous strategies.

In summary we propose the following hub selection strategies:

- 1-neighborhood median: A node is selected as a hub if its hub value is greater than the median of hub values in its one-neighborhood.
- 1-neighborhood mean with multiplier: A node is selected as a hub if its hub value is greater than the mean of hub values in its one-neighborhood multiplied with a parameter $q > 0$.
- 1-neighborhood weighted mean with multiplier: The hub values are multiplied with the strength of the nodes. Beside this, the strategy is the same as above.

As a recommendation, the median strategy should be tried first, and if it does not give satisfactory results the average strategy should be used with $q = 1$ initially, decreasing or increasing its value in small steps depending on the requirements. In practice $0 < q < 2$ seems to hold.

3.2 Implementation

The bottleneck of the algorithm is finding all maximal cliques in graph G. A general graph with n vertices may contain up to $3^{n/3}$ maximal cliques. In correspondence, the original algorithm of Bron and Kerbosch has a worst-case running time of $O(3^{n/3})$. In small-world networks however, the number of maximal cliques is smaller by magnitudes, decreasing the running time of the algorithm. Furthermore, refinements of the Bron-Kerbosch algorithm have been published in recent years, enabling the use of this method on large sparse networks [30, 8]. In cases when even faster computation is required, there are existing heuristics for clique search [5].

The hub value of each node can be calculated in a single pass on the set C of cliques. All of the hub selection strategies suggested in the previous section have a local fashion: they can be computed in a single pass on the vertices and their one-neighborhoods.

The computation of C_H does not require a repeated run of the Bron-Kerbosch algorithm on $G[H]$, since the cliques of G contain the cliques of $G[H]$ as subsets. Therefore it is enough, that for each $c \in C$, if $|c \cap H| \geq k$, simply add all k-combinations of c to C_H. Depending on the size of the network and the hub selection strategy, H may be quite large, but the use of flags on the nodes of the graph G to signal the hub property can reduce the computation of this step to a single pass on C. The percolation step can be executed by computing the 1-neighborhood of each $c_H \in C_H$. Let c_H^+ contain all of the direct neighbors of vertex set c_H, and initially let $c_e \leftarrow c_H$. For all nodes $v \in c_H^+$, if $|\{v\}^+ \cap c_H| \geq 2$ add v to c_e. Again this step can be computed by making a single pass on C_H.

[5] The median hub selection strategy was used with $k = 2$, see subsection 3.1.

[6] The sum of the weights on all adjacent edges.

In order to make the joining step, the computation of the hubs of each elementary community is required: for each $c_e \in C_e$ let $c_{H_e} = c_e \cap H$. Let C_{H_e} denote the sets of hubs of the elementary communities. An important remark is, that $C_{H_e} \neq C_H$ since in the previous step additional hubs may have been added to the elements of C_H. Removing the ”sub-hubs” (hubs being contained in other elements of C_{H_e}) can be executed in quadratic time in worst case. In general, performance can be improved by sorting C_{H_e} in descending order according to the sizes of $c_{H_e} \in C_{H_e}$. After this, starting from the first element, remove all the sets of vertices from C_{H_e} which are subsets of the first one, then repeat for the second, third, ... until no more vertex sets can be removed from C_{H_e}. Finally, the elements of C_e and C_{H_e} must be compared: for all $c_{H_e} \in C_{H_e}$ find all $c_e \in C_e$ where $c_e \cap H \subseteq c_{H_e}$ and take the union of these vertex sets.

We can conclude, that the two most time-consuming steps of the method is the computation of C and C_H, all other operations take at most quadratic time[7]. The algorithm of Eppstein and Strash [8] is able to list all maximal cliques in large sparse networks in reasonable time. For faster computation heuristics [5] or the use of quasi-cliques [23] can be applied. The size of C_H depends on two factors: the size of H and k. The former is governed by the hub selection strategy, the latter is a parameter of the algorithm. Choosing a different hub selection strategy, that produces a smaller number of hubs, or decreasing k may speed up computations.

4 Sensitivity to parameters

We have created the hub percolation method with the intent to provide a versatile tool for community detection. Therefore, an important question arises: how does the hub selection strategy and the filtering parameter influence the community structure found by the algorithm? For the purpose of examining their effect, we will use several well-known benchmark networks including the word association graph of Palla et al.[29], a scientific collaboration network [26] and a graph of American football games [11].

The first network we will examine was created by Newman [26] on the condensed matter archive at www.arxiv.org based on preprints posted to the archive between January 1, 1995 and March 31, 2005. The graph is undirected, unweighted and contains 39540 nodes and 175683 edges. We will evaluate the median and average hub selection strategies and we will also experiment with different values for k. We will measure the number of communities, the average overlap[8], the number of singletons[9] and hubs in the network. We will also present the community size distribution for each selection strategy.

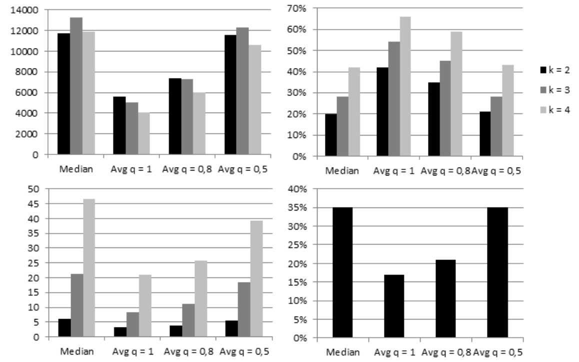

Figure 2: Upper left: Number of communities for different hub selection strategies and values for k; Upper right: The percentage of nodes without communities; Lower left: The average overlap; Lower right: The percentage of hub nodes in the network.

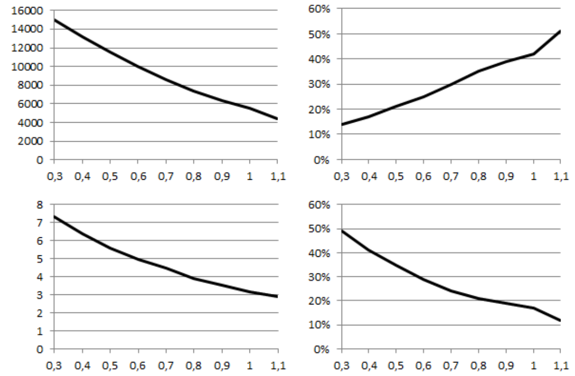

Figure 3: Upper left: Number of communities for the average hub selection strategy with $q = 0.3, \ldots, 1.1$ and $k = 2$; Upper right: The percentage of nodes without communities; Lower left: The average overlap; Lower right: The percentage of hubs in the network.

On Figure 2 we have compared four different hub selection properties: the median strategy and the average strategy with values $q = 1, 0.8, 0.5$. The number of communities is the greatest and the number of singletons is the lowest with the median strategy and the average strategy with $q = 0.5$; these strategies provide the greatest cover on the network. The number of hubs is also the greatest with these strategies: roughly one in three nodes, this confirms our expectations, that hubs should be ”common”. We can see, that the average overlap and the number of singletons increases with k, while the number of communities does not change. The reason for the above fact is that by increasing k, the nodes are concentrated in highly overlapping communities keeping the number of communities constant, while many nodes are left out of the community building process.

We will further examine the average hub selection strategy with $k = 2$ on Figure 3. The main observations remain the same with higher values for k. As before, the number of hubs and communities grows inversely proportional

[7] We have also conducted experiments, and found that clique detection may take from 60% up to 95% of the running time.

[8] The sum of the cardinalities of all community divided by the number of nodes.

[9] Nodes without communities.

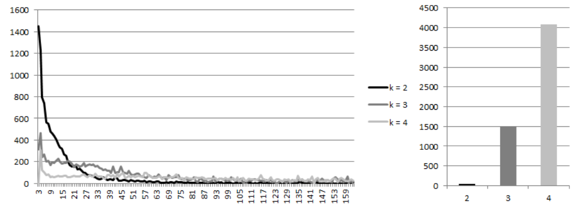

Figure 4: The community size distribution of the median hub selection strategy with different values of k. Left: The number of communities with size below 150. Right: The number of communities with size greater than 150.

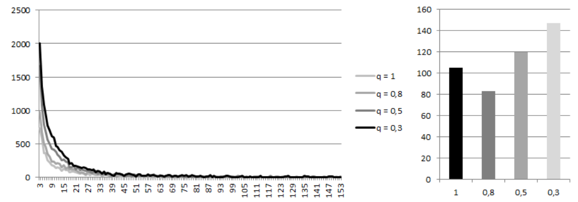

Figure 5: The community size distribution of the average hub selection strategy with different values of q, $k = 2$. Left: The number of communities with size below 150. Right: The number of communities with size above 150.

with q, while the number of singletons grows proportionally with it. The average overlap slowly decreases when q is increased, indicating that decreasing the number of hubs causes communities to become smaller and scarcer.

We can see, that the community size distributions follow a power-law on Figures 4 and 5. The median hub selection strategy is depicted with different values for k. Increasing k results in much larger communities: With $k = 2$, The largest community had 255 members, with $k = 4$ the maximum was 869. This confirms our previous observation, that increasing k creates a highly overlapping community structure. Similar observations can be made with the average hub selection strategy. Increasing q decreases the number of communities evenly among the community sizes, even the size of the largest communities does not change much.

A strict requirement for all community detection algorithms should be, that the number of nodes left without community memberships should be minimized. Therefore we can conclude, that the filtering parameter should be kept as low as possible, and the ratio of hubs should be above 30%.

We have measured the running time of our method as well[10]. The results for the average hub selection strategy with different values of q and k can be seen on Figure 6. We have seen before, that decreasing q increases the number of hubs – the size of H and C_H. This directly increases the computational time of the joining phase. The filtering parameter k also has an impact on the running time of the method, since it influences the size of C_H. As a conclusion we can say, that the filtering parameter should be kept as low as possible, and the average hub selection strategy should be used to further refine the results of the algorithm.

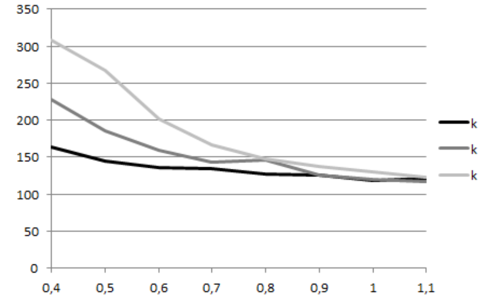

Figure 6: The running time measured in seconds with the average hub selection strategy and different values of q and k.

We can draw similar conclusions on the other two networks, with a few exceptions. The relationship between the ratio of hubs, the community size and the average overlap is the same in all networks. The ratio of singletons shows a similar behavior as it grows inversely proportional to the ratio of hubs. There is a difference however; the graph of football games contains no singletons for the majority of the parameter configurations, while the ratio of singletons never goes below 30% in the word association network. This can be explained by the difference in the structure of the networks. The graph of football games is an union of cliques by definition, while word associations do not have this property. Since our method is clique-based, it is able to cover all nodes of the former test set, while in the latter case nodes not part of any triangles are left out of the building process.

The relationship between the ration of hubs and the hub selection strategies is also similar, that is for the average selection strategy increasing q decreases the number of hubs. However, the exact pairs of these values change together with the networks. For example setting $q = 0.5$ results in 35% of nodes being selected as hubs on the collaboration network, 21% on the word association network and 90% on the graph of football games. Therefore in any application, it is important to find the hub selection strategy that produces the ratio of hubs so that the number of communities, the size of the overlaps and the number of singletons move according to the specifications of the application.

We have previously concluded that the filtering parameter should be kept as low as possible to reduce both the ratio of singletons and the computation speed. As we will see below, there are some situations where a higher value is desirable. In the next chapter we are going to examine networks with a large number of highly overlapping communities.

5 Performance on benchmark networks

For the purpose of evaluation, we have used benchmark networks created with the graph generator of Lancichinetti

[10] We have implemented our method in JAVA, and we have used a computer with an Intel i7-2630QM processor, and 8 gigabytes of memory.

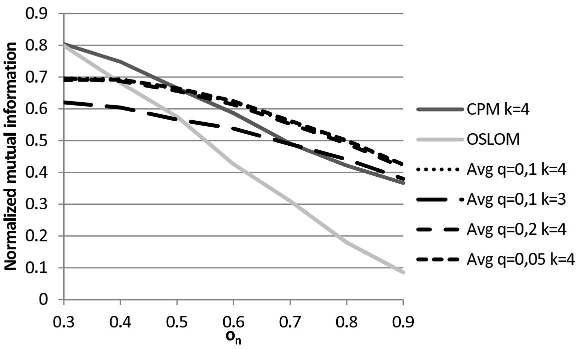

Figure 7: The performance of hub percolation compared to CPM and OSLOM with $\mu_t = 0.1$.

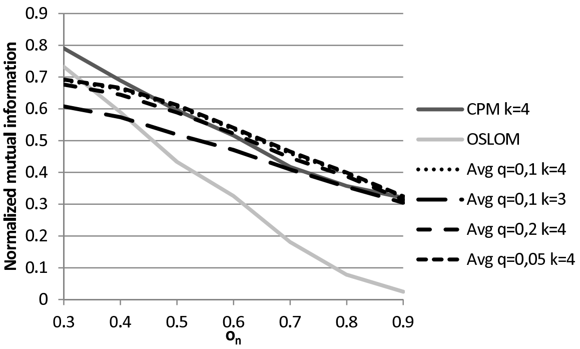

Figure 8: The performance of hub percolation compared to CPM and OSLOM with $\mu_t = 0.2$.

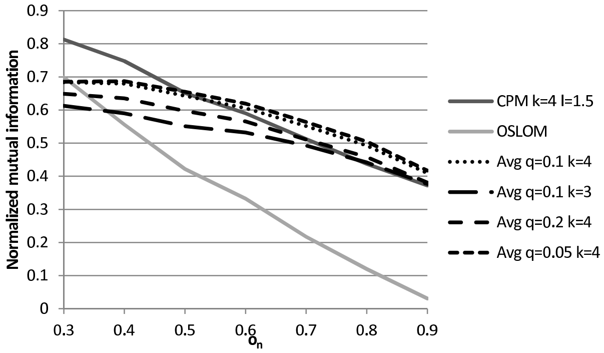

Figure 9: The performance of hub percolation compared to CPM and OSLOM with $\mu_t = \mu_w = 0.1$.

and Fortunato [18]. We have generated both weighted and unweighted networks, with the following parameters:

- We have created undirected graphs with $|V(G)| = 1000$
- The average degree was 15
- The maximum degree was 50
- The exponent of the degree distribution was -2
- The minimum community size was 3
- The maximum community size was 25
- The exponent of the community size distribution was -1
- The mixing parameter μ_t was between 0.1 and 0.2
- The fraction of overlapping nodes o_n was between 0.3 and 0.9

A detailed description of the used model and its parameters can be found in [18]. We have selected the parameters above, because they are close to the recommendations of Lancichinetti and Fortunato, yet they provide a challenge to our method. Again following the recommendations of the above authors, we have used mutual information [19] to measure the similarity between the communities given by our method and those of the benchmark. Because of the probabilistic nature of the benchmark we have generated 10 different networks for each parameter configuration and averaged the similarity measurements.

We have compared the performance of our method to that of the clique percolation method and OSLOM. We have tried several values for k-clique percolation, and have found that $k = 4$ clearly provides the best results, therefore we have used this parameter setting for comparison. We have also made comparisons with COPRA but found, that the above methods are clearly superior on these benchmark networks, so we have omitted these results from the figures.

On Figures 7 and 8 we can see that the best results were provided by the 1-neighborhood average hub selection strategy with low q values and $k = 4$. We can also see, that our method reaches peak performance at $q = 0.1$, but the selection of q has little influence on the results. The median selection strategy performs poorly on these networks, and increasing or decreasing k worsens performance. If we compare our method to CPM and OSLOM, we can conclude, that hub percolation gives better results on networks with a high number of overlapping nodes.

Our observations remain the same when using the weighted benchmarks of the same authors with the recommended parameters $\mu_t = \mu_w$ and $\beta = 1.5$. Low values of q and $k = 4$ gives the best results for hub percolation, and 4-clique percolation with a weight threshold $l = 1.5$ is the best for CPM. As before, hub percolation gives better results on networks with a high number of overlapping nodes.

6 Case-study: an economic network

In this section, we will examine the community structure of a specific economic network constructed from the Hungarian company register. We will consider a network of companies: each vertex is a special type of company (Ltd.), and the companies are connected if they share a common owner (or member in the case of Ltd.'s). We will call this network as an intersection network, because two vertices

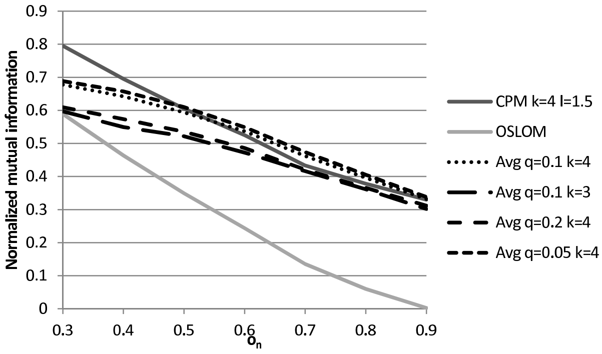

Figure 10: The performance of hub percolation compared to CPM and OSLOM with $\mu_t = \mu_w = 0.2$.

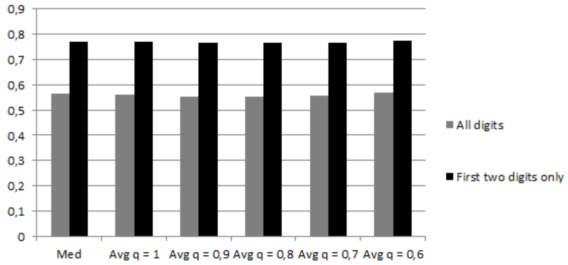

Figure 11: The locality of the communities of the intersection network with different hub-selection strategies, using all digits of the zip-code (left) or the first two digits (right).

are connected, if the sets of owners associated with them have a non-empty intersection. Due to the changes in the regulations governing the company register's construction, there are large amounts of missing and erroneous data. The register's sometimes has unordered structure so the identification of the companies, owners and the construction of the graph itself required the application of several data mining methods, data cleaning and filtering.

The resulting graph is not connected, there is a high number of small disconnected components in it, but fortunately it contains a giant component as well. The small components often cannot be divided into two or more communities, thus they do not provide useful information about the structure of the graph. Therefore, in our analysis we will consider only the giant component. This graph is a small-world network with the previously mentioned properties. It has 239685 vertices and 1423080 edges. Depending on the hub-selection strategy, our method was able to discover the community structure of this network in 5-7 hours[11]. We have considered comparing our method with CPM on this dataset, but the publicly available[12] implementation was unable to produce results.

There are several points of interests regarding the community structure of the network. In this paper, we are going to focus on three of them. The first one is the geographical location of these groups and companies inside them. Our main question is, are the communities of the graph local in a geographical sense? Using the register, we can assign zip-codes to the companies, and by counting the number of different zip-codes inside the community – the frequency of individual zip-codes, we can easily address the above question. We can further divide the frequency of the most frequent location by the size of the community, and by averaging this fraction over all of the communities we can represent the locality of these communities as a simple number. The structure of the zip code also allows us to fine-tune the resolution of the analysis. The Hungarian zip-code contains four digits: the first one divides the country into nine large regions, the first two identifies 80 sub-regions. On Figure 11 we can see the computed average locality of the communities. Both the accurate locations – all digits of the zip-code – and the sub-region classification system is used. We can conclude, that the communities are local indeed; in average 77% of companies inside communities belong to the same sub-region, and even in the case of the accurate locations, this percentage does not go below 55%. This implies, than companies owned by the same people tend to stay in the same geographical area. It is important to emphasize, that we are observing a special form of companies: the Ltd.'s. Our results makes sense, because this company form is popular for small companies, that do not have the resources to cover a large area. On Figure 11 we can also see a comparison between the different hub-selection strategies[13]. Even though the number of communities and the size of overlap changes according to the observations in the previous section, all strategies gave a similar stable performance.

We can perform the same analysis considering the industrial sectors the companies belong to. Do the communities of the graph belong to similar industrial sectors? The sector classification numbers for the individual companies are available, but due to changes in regulation it is impossible perform a high-resolution scan. On the other hand we can make use of a rough classification system containing 118 different industrial sectors. The method is the same as before: we compute the most frequent sector for each community, and we average the relative frequencies over all communities. As a result we can say, that in average 84% of the companies inside the communities belong to the same industrial sector. The communities are even more "local" to the industrial sector most of their members belong to, than to their geographical location. The reason for this is similar to the previous one: small companies tend to specialize, and it is rare for an owner to have an interest in multiple sectors. Again, we have compared different hub-selection strategies and found, that they have similar performance.

We can see a small example of this behavior on Figure 12. The whole economic graph is too large to visualize, so we are going to take a look at a small subgraph of three communities. The red community contains companies fo-

[11] On the same hardware as above.

[12] We have used CFinder [29] downloaded from http://cfinder.org/

[13] $k = 2$ was used in all experiments.

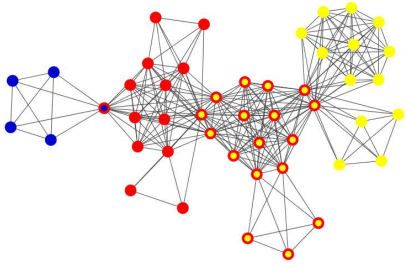

Figure 12: Three communities of the economic intersection graph.

cusing on printing services, the blue one on distributing in general, while the companies in the yellow one are centered around public utilities in real estate and engineering. A huge overlap can be seen between the red and yellow companies: vertices in the non-overlapping part of the red groups are focused on distributing, while the companies in the overlap are either copy shops, smaller publishing companies or hardware and electronics retail shops.

Our last point of interest is the age of the companies. Since the date of establishment is available for all companies, we can ask the question: Were the companies inside the communities of the graph established in a short time period? We can answer this question by computing the standard deviation of the dates of establishments for all companies. The expected value of the standard deviation inside the communities is 5 years. This relatively large value indicates, that the establishment of these companies is spread in time over a considerable interval.

As a conclusion we can say, that our method is capable of identifying communities that share a common geographical location and industrial sector.

7 Conclusions

In this paper, we have introduced the hub-percolation method: a clique-based high-resolution overlapping community detection algorithm. This method is based on two observations: cliques are the most natural representations of communities, and some vertices are crucial in the birth of communities: they connect different sub-communities together by forming bridges between them. There are multiple ways to select these vertices; that authors have suggested several hub-selection strategies, some of them have tunable parameters. The method also has a filtering parameter k which influences the size and structure of the overlaps between the communities. Adjusting k and the hub-selection strategy allows the user to apply this method to a variety of small-world networks with different densities. It also allows the user to discover several layers of community structure on the same network. We have examined the effect of different parameter choices on several well-known benchmark networks. We have concluded, that the selection strategy should be chosen so that 30-50% of the vertices are selected as hubs and k should be kept low to minimize the number of singletons.

We have shown two ways to measure the goodness of the hub-percolation algorithm. One of them was an economical case-study, a network where the vertices represent companies, or more precisely Ltd.'s, and the companies are connected if they share one or more members. Our method is able to identify communities, that are geographically local and belong to the same industrial sectors in reasonable time considering the size of the network. We have also used benchmark graphs created with the graph generator of Lancichinetti and Fortunato [18]. Using these networks, we have compared our method with the well-known clique percolation algorithm of Palla et al. We have concluded, that in average the two methods have similar performance, but hub-percolation gives better performance on networks with a high number of overlapping nodes.

Finally, a slight adjustment in the hub-selection strategy allows us to handle weighted networks without the need to filter the graph edges according to some possibly non-trivial weight limit. Using the previously mentioned graph generator, we have created weighted benchmark graph, and compared the goodness of our method with those of the weighted clique percolation algorithm. Our conclusions were the same as before: similar performance in average, better results with high overlaps.

Acknowledgement

The first author was supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013)

The second author was supported by the European Union and co-funded by the European Social Fund. Project title: “Telemedicine-focused research activities on the field of Mathematics, Informatics and Medical sciences.” Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073 .

References

[1] Y-Y. Ahn, J. P. Bagrow, S. Lehman, Link communities reveal multiscale complexity in networks. *Nature*, **466**(7307):761–764, 2010.

[2] A-L. Barabási, R. Albert, Emergence of Scaling in Random Networks. *Science*, **286**(5439):509–512,1999.

[3] A. Bóta, L. Csizmadia, A. Pluhár, Community detection and its use in Real Graphs. *Proceedings of the 2010 Mini-Conference on Applied Theoretical Computer Science*, 95–99, 2010.

[4] F. Cazals, C. Karande, A note on the problem of reporting maximal cliques, *Theoretical Computer Science*, **407**(1):564–568, 2008.

[5] James Cheng, Linhong Zhu, Yiping Ke, and Shumo Chu, Fast algorithms for maximal clique enumeration with limited memory. *Proceedings of the 18th ACM SIGKDD*. ACM, New York, 1240–1248, 2012.

[6] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár, T. Tóth, The use of infection models in accounting and crediting. *Challenges for Analysis of the Economy, the Businesses, and Social Progress* Szeged (2009) pp. 617–623..

[7] A. Decelle, F. Krzakla, C. Moore, L, Zdeborova, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, **84**(6):066106, 2011.

[8] D. Eppstein, D. Strash, Listing all maximal cliques in large sparse real-world graphs. *Experimental Algorithms*, Springer Berlin Heidelberg, 364–375, 2011.

[9] T. Evans, R. Lambiotte, Line Graphs, Link Partitions and Overlapping Communities, *Phys. Rev. E*, **80**(2):016105, 2009.

[10] S. Fortunato, Community detection in graphs. *Physics Report*, **486**(3):75–174, 2010.

[11] M. Girvan, M. E. J. Newman, Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, **99**(12):7821–7826, 2002.

[12] P. K. Gopalan, D. M. Blei, Efficient discovery of overlapping communities in massive networks. *PNAS*, **110**(36):14534-14539, 2013.

[13] S. Gregory, Finding overlapping communities in networks by label propagation. *New J. Phys.*, **12**(10):103018, 2010.

[14] E. Griechisch, A. Pluhár, Community Detection by using the Extended Modularity. *Acta Cybernetica*, **10**:69–85, 2011.

[15] D. Kempe, J. Kleinberg, E. Tardos, Influential Nodes in a Diffusion Model for Social Networks. *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, Springer-Verlag (2005) 1127–1138.

[16] M. Krész and A. Pluhár, Economic Network Analysis based on Infection Models. To appear in *Encyclopedia of Social Network Analysis and Mining*, Springer (2014).

[17] J. M. Kumpula, M. Kivela, K. Kaski, J. Saramaki, Sequential algorithm for fast clique percolation. *Phys. Rev. E*, **78**(2):026109, 2008.

[18] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, **80**(1):016118, 2009.

[19] A. Lancichinetti, S. Fortunato, J. Kertész Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, **11**(3):033015, 2009.

[20] A. Lancichietti, F. Radicchi, J. J. Ramasco, S. Fortunato, Finding statistically significant communities in networks. *PLoS One*, **6**(4):e18961, 2011.

[21] C. Lee, F. Reed, A. McDaid, N. Hurley, Detecting highly overlapping community structure by greedy clique expansion. *Preprint*, 2010.

[22] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, **54**(4):396-405, 2003.

[23] V. Maniezzo, R. Battiti, J-P Watson, On Effectively Finding Maximal Quasi-cliques in Graphs. In *Learning and Intelligent Optimization*, Lecture Notes in Computer Science (5313) 41–55, Springer Berlin Heidelberg, 2008.

[24] J. Moon, L. Moser, On cliques in graphs. *Israel Journal of Mathematics*, **3**(1):23-28, 1965.

[25] T. Népusz, A. Petróczi, L. Négyessy, F. Bazsó, Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E*, **77**(1):016107, 2008.

[26] M. E. J. Newman, The structure of scientific collaboration networks. *PNAS*, **98**(2):404–409, 2001.

[27] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks. *Phys. Rev. E*, **69**(2):026113, 2004.

[28] V. Nicosia, G. Mangioni, C. Carchiolo, M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, **3**:P03024, 2009.

[29] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**(7043):814–818, 2005.

[30] E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, **363**(1):28–42, 2006.

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

[32] W. W. Zachary, An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 452–473, 1977.