Community detection by graph Voronoi diagrams

Dávid Deritei¹,³, Zsolt I Lázár¹,³, István Papp¹,³, Ferenc Járai-Szabó¹, Róbert Sumi¹, Levente Varga¹, Erzsébet Ravasz Regan² and Mária Ercsey-Ravasz¹

¹ Faculty of Physics, Babeș-Bolyai University, Str. M.Kogălniceanu 1, RO-400084 Cluj-Napoca, Romania
² Center for Vascular Biology Research, Department of Medicine, Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, USA
E-mail: ercsey.ravasz@phys.ubbcluj.ro

Received 4 January 2014, revised 10 April 2014
Accepted for publication 24 April 2014
Published 5 June 2014

New Journal of Physics 16 (2014) 063007
doi:10.1088/1367-2630/16/6/063007

Abstract
Accurate and efficient community detection in networks is a key challenge for complex network theory and its applications. The problem is analogous to cluster analysis in data mining, a field rich in metric space-based methods. Common to these methods is a geometric, distance-based definition of clusters or communities. Here we propose a new geometric approach to graph community detection based on graph Voronoi diagrams. Our method serves as proof of principle that the definition of appropriate distance metrics on graphs can bring a rich set of metric space-based clustering methods to network science. We employ a simple edge metric that reflects the intra- or inter-community character of edges, and a graph density-based rule to identify seed nodes of Voronoi cells. Our algorithm outperforms most network community detection methods applicable to large networks on benchmark as well as real-world networks. In addition to offering a computationally efficient alternative for community detection, our method opens new avenues for adapting a wide range of data mining algorithms to complex networks from the class of centroid- and density-based clustering methods.

³ Equal contribution.
Keywords: complex networks, community detection, Voronoi diagrams

1. Introduction

Network representations of complex systems are now common in social science, information theory, technology, biology and neuroscience, to name a few [1, 2]. A common challenge across all of these approaches is the identification of the community structure in complex networks. In general, communities are defined as groups of nodes which are more densely connected to each other than with the rest of the network. However, this is a qualitative description and no common mathematical definition has been agreed upon [3], as illustrated by an entire subclass of methods that work with operational definitions of network community (see below). While a large variety of community definitions and detection algorithms exist (for an extensive review, see [3]), combining meaningful mathematical community definitions with computationally efficient methods remains a challenge.

One class of methods formulates community detection as a global optimization problem. In these methods the quality characterizing a particular graph partitioning is measured by a fitness function (e.g., modularity), and communities are found by maximizing this quality function [3–6]. However, finding the optimum partitioning is an NP-hard problem, thus extremely inefficient on large networks. This difficulty led to the development of approximations based on local optimization [7]. A second class of algorithms is characterized by operational definitions of community structure. They prescribe a procedure or dynamics that partition the graph, and the partitioning is accepted as its community structure. For example, divisive/aggglomerative algorithms consist of an iterative process of link removal/inclusion based on some link property (e.g., betweenness centrality [8] or similarity in link-community detection [9]), resulting in a dendrogram. While these methods are typically more efficient than optimization-based clustering, it is usually not obvious where the resulting dendrogram needs to be cut to obtain meaningful partitioning. This problem was mitigated by the formulation of a validity check for communities, making the cutting of dendrograms more reliable [10]. Dynamic processes on complex networks inspired a set of algorithms based on stochastic processes such as random walks [11] and label propagation [12]. In this case, besides choosing between the large spectrum of proposed dynamics, one must also deal with high sensitivity to parameters [3]. In summary, the strong dependence of the efficiency and accuracy of these methods on the particularities of the analyzed network have so far prevented the emergence of a single canonized community detection procedure.

Clustering problems similar to graph community detection also occur in data mining, pattern recognition, machine learning and statistical data analysis [13, 14]. These problems, however, are defined on continuous metric spaces, simplifying their formulation. Voronoi diagrams [15] are a common way of partitioning metric spaces into regions (Voronoi cells) around a given set of points called seeds (or generator points), such that each point of the space belongs to the cell of the closest seed (figure 1(A)). $k$-means clustering, a well-known NP-hard problem in data mining [16], partitions a set of data points (observations) into $k$ Voronoi cells. To do this, the method identifies a set of seeds for which the average square distance (e.g., Euclidean distance) between data points and the Voronoi seeds they belong to is minimal.
Although several attempts to map graphs to metric spaces have been reported, they did not yield a general, computationally efficient network community detection algorithm. For example, Nishikawa et al defined a multidimensional metric space of arbitrarily chosen node properties, such as the average degree of first neighbors or the clustering coefficient [17]. They then projected this space onto two dimensions and inspected them visually to discover group structures. It is not clear whether this method identifies structural network communities, or node groups that occupy structurally similar positions in the network. Jin et al used density-based clustering techniques to define a distance function between nodes [18]. However, their distance calculation is a multistep procedure involving global optimization. Combined with a sophisticated clustering algorithm, the procedure is of high-order time complexity, $O(n^\alpha)$, $\alpha > 2$.

A straightforward alternative of mapping graphs onto metric spaces (although not continuous) states that all edges have a positive length and the distance between two nodes is the length of the corresponding shortest path [19]. In this paper we expand on this alternative, and define network communities as graph Voronoi cells [20] (figure 1(B)). We argue that this simple geometrical approach provides reliable results for a broad range of networks at low computational cost.

Our approach raises two major challenges. First, we need a meaningful definition for the length of the edges—one that efficiently separates communities in the metric space. A variety of widely used local topology-based similarity/dissimilarity measures meet this criterion [3]. We chose one of the simplest measures, namely the inverse of the edge-clustering coefficient [10]. This measure assures that links connecting different clusters are longer than intra-community links. While more complex definitions are certainly possible, we have found this simple approach to provide very good results. The second challenge is to identify Voronoi seed nodes based on topological information. In contrast to $k$-means clustering, we do not $a$ priori specify the number of communities. As Voronoi seeds need to be in the ‘center’ of communities, we propose a density-based approach. We identify seeds as local maxima in the ‘density landscape’

![Figure 1. Voronoi diagrams. (A) Illustration of Voronoi partitioning in 2D Euclidean space. (B) Graph Voronoi diagram as represented by the graph drawing application Gephi using the ForceAtlas2 layout algorithm [21]. Generator nodes are shown in black.](image)
of the graph. Visually, one can imagine that each node has a height proportional to its relative local density [3]. Communities appear as hills in this landscape, while generator nodes are those at the top of the hills (see figure 2(B)).

The resulting method is computationally efficient. The complexity of graph Voronoi partitioning is $O(N \log N)$ [20], where $N$ is the number of nodes. Both the edge clustering coefficients and relative local densities can be calculated in $O\left(M \langle k \rangle^2 \right)$ time, where $M$ is the number of edges and $\langle k \rangle$ is the average degree. Lastly, finding the Voronoi seeds can be achieved well below $O(Mg)$, where $g$ denotes the number of communities (and thus the number of generator nodes). We tested the method on benchmark networks as well as real datasets, such as the scientific collaboration network of ArXiv [22], the protein interaction network of yeast [23], the political blog network [24], the neural network of C. elegans [25, 26], as well as the Zachary karate club [27]. Although we concentrated on partitioning graphs into non-overlapping communities, at the end of the paper we offer an outlook of how the method could be generalized to fuzzy clustering.

2. Illustration in 2D Euclidean space

In order to introduce the logic of our clustering method, we first illustrate its essentials on a clustering problem defined in Euclidean space. Imagine a set of points thrown onto a 2D plane,
unevenly distributed to form local groups (figure 2(A), black dots). The question is, how can we partition the space to separate these clusters? First, we define a square lattice across the plane, and calculate the local density in each small square as the number of black dots included in it. For example, the space in figure 2(A) is partitioned into 12 × 25 squares and colored based on local density (from green/low to red/high). Second, we choose squares with the largest density inside their neighborhood with radius $r$ as Voronoi cell seeds (generator points). In the top image in figure 2(A), we chose $r = 4$ units, yielding 7 seeds (white stars; gray circles indicate their local neighborhood). Third, we assign each point of the plane to a Voronoi cell belonging to the seed closest to the point. This yields a Voronoi partitioning diagram, where partition boundaries are equidistant lines to at least two Voronoi seeds (blue lines). The resulting Voronoi cells partition the set of black dots into groups which correspond well to communities identifiable with the naked eye. Increasing the radius leads to the sequential disappearance of seeds, resulting in larger clusters. This process typically merges whole clusters, without splitting them in meaningless ways. For example, increasing the radius from $r = 4$ to $r = 7$ (figure 2(A)) merges smaller clusters and partitions the same set of black points into two clusters. This suggests that varying the parameter $r$ may be leveraged to detect structural hierarchy.

Occasionally, two squares closer than $r$ happen to have the same local density. In this case, the Voronoi seed is chosen randomly from the two candidates. This rarely occurs in large systems (especially on graphs; see below). Moreover, the algorithm is not sensitive to the above random choice, as illustrated in figure 2(A), top panel. Here, the squares with coordinates $(8, 7)$ and $(9, 8)$ have the same density. In the figure we chose $(9, 8)$ as the seed, but choosing the other one only changes the community association of the two black dots ($(7, 4)$ and $(9, 12)$). Both dots hang at the edge of their cluster in both partitioning results, not being strongly associated with either cluster.

In addition to the radius $r$, the method applied to this Euclidean toy model depends on the lattice cell size used to calculate local density. On graphs, a more natural definition for local density exists that does not require an initial lattice. We emphasize that the above method is presented for the sole purpose of visually illustrating the elements of our proposed graph community detection method. As clustering points in Euclidean space is a central issue in data mining, efficient methods abound [13, 14]. However, most of these algorithms cannot be directly applied to graphs.

3. Generalization of the method on graphs

3.1. Graph Voronoi diagrams

Let $G = (V, E)$ be a directed and weighted graph with the set $V$ of $N$ vertices (nodes) and set $E$ of $M$ edges (links). We denote the length (weight) of an edge connecting nodes $\nu$ and $\mu$ as $l(\nu, \mu) > 0$. The length of a path is obtained by summing up the length of edges constituting the path. The distance $d(\nu_i, \nu_j)$ between two nodes $\nu_i$ and $\nu_j$ is the length of the shortest path between the two nodes. Thus, the definition of an edge length assures that the graph can be treated as a metric space. The simplest choice of $l(\nu, \mu)$ is 1 if $\nu$ and $\mu$ are directly connected, 0 otherwise.
Let us now choose a set \( S \equiv \{\gamma_1, \gamma_2, \ldots, \gamma_g\} \subset V \) of generator points or seeds. The Voronoi diagram of graph \( G \) with respect to \( S \) is thus the partitioning of \( V \) into node sets \( V_1, V_2, \ldots, V_g \subset V \), where each point set or Voronoi cell is associated with a generator point and

1. The Voronoi cells cover the original graph with no overlaps:
\[
\bigcup_{i=1}^{g} V_i = V, \quad V_i \cap V_j = \emptyset, \quad i, j = 1, 2, \ldots, g.
\]

2. Nodes that belong to a Voronoi cell are closest to the generator point associated with the cell:
\[
d\left(\nu, \gamma_i\right) \leq d\left(\nu, \gamma_j\right), \quad \text{for all } \nu \in V_i, \quad i, j = 1, 2, \ldots, g.
\]

Mathematical properties of graph Voronoi diagrams, along with different methods for their identification and their respective computational complexity, are detailed in [15].

To illustrate the potential of the Voronoi partitioning method for community detection, we first applied its crudest version to an extremely modular benchmark network with predefined communities (figure 1(B)). Here, we selected an arbitrary generator point from each known community (black nodes), and simply assumed that all edges have unit length. The ForceAtlas2 layout algorithm in the Gephi drawing application [21] aptly coalesces the sharply separable clusters of this network. We then added node color to indicate Voronoi cell membership. The near-perfect mapping between Voronoi cells and the community structure visualized by the layout algorithm suggests a low sensitivity of Voronoi partitioning to the choice of generator nodes and of the distance measure (definition of the length of edges). Next, we show that identification of a meaningful distance measure and optimal generator points leads to excellent results even on more complex graphs.

### 3.2. Distance measure

For optimal use of our geometrical approach, we need a distance measure definition which translates node membership in different communities into separation in metric space. In other words, the more likely an edge connects two different clusters, the longer it should be. There are many distance measures which qualify for this purpose (see discussion). Here we chose to adapt one of the simplest, well accepted and computationally cost-effective measures, namely the edge clustering coefficient (ECC) introduced in [10]. The ECC of an edge connecting node \( i \) to node \( j \) is defined as

\[
C_{ij} = \frac{z_{ij} + 1}{\min \left[ (k_i - 1), (k_j - 1) \right]}
\]

where \( k_i, k_j \) are the degrees of the two nodes, \( z_{ij} \) is the number of triangles the edge belongs to, and \( \min \left[ (k_i - 1), (k_j - 1) \right] \) is the number of potential triangles it could belong to, as it is the smaller value of the degrees of the two adjacent nodes, minus one (the examined edge). This measure characterizes the probability that an edge connects nodes of different clusters (inter-cluster) or nodes in the same cluster (intra-cluster). The smaller the ECC, the more likely an
edge is to connect nodes in different clusters. Consequently, we define edge length (weight) as the inverse of ECC in our graph Voronoi clustering method. The length of any path in the graph is then the sum of the length of edges along the path, and the distance between two nodes is the length of the shortest path.

### 3.3. Identification of generator points

In addition to an appropriate distance measure, optimal choice of generator nodes can further improve the match between network communities and graph Voronoi cells. A principal goal is to select one central node in each cluster. In the rare case when one has a conjecture or a sample of the community structure, random nodes belonging to supposedly different communities can work as generator points (as illustrated in figure 1(B)). For the general case, however, no such information or metadata is available. Here we offer a Voronoi seed selection method based on the relative local density of nodes [3], which is calculated as:

$$ \rho_i = \frac{m}{m + k} $$

where $m$ is the number of edges inside the subgraph consisting of the first neighbors of node $i$, and $k$ is the number of edges going out of the neighborhood. The value of this density is highest for nodes within the center (dense parts) of communities, as illustrated in figure 2(B).

As in Euclidean space, generator points on a graph will be chosen as the nodes with the highest local density in a region within radius $r$. On graphs, $r$ is measured using the above defined distances between nodes. This method for selecting community centers avoids the selection of multiple generator points within one cluster. In the case when two nodes have equal density we choose one of them randomly (this rarely happens in large networks; see the supporting information, figure S1, available at stacks.iop.org/njp/16/063007/mmedia).

Varying the parameter $r$ leads to partitioning with a different number of clusters and can reveal a hierarchical community structure within the graph. As illustrated in Euclidean space, the algorithm is not sensitive to $r$. Cluster memberships obtained at similar $r$ values are near-perfectly matched (figures S2(A),(B)), with the exception of $r$ values where cluster mergers take place in hierarchical networks (figures S2(C)–(E)). As we will show below in benchmark and real networks, extremely small $r$ values should not be considered, but relatively small $r$ values already yield good partitioning, keeping the algorithm computationally efficient. We propose that the best strategy is to partition the graph with increasing values of $r$, while monitoring a quality function of the resulting clustering (e.g., modularity, separability). Optimum quality is usually achieved at small values of $r$, as shown below.

### 3.4. Computational efficiency

The algorithm has three main parts. (i) First, we calculate the local density of each node and the ECC of each link, achievable in $O(N \langle k \rangle^2)$ and $O(M \langle k \rangle^2)$ steps, respectively. (ii) Second, we identify the generator points by checking whether a node has the highest local density within a radius $r$. A breadth-first search covering the whole network from one node has complexity $O(M)$, but we find that searching at depth is not necessary for optimal partitioning. Our results show that large values of modularity are achieved at small values of the radius parameter (typically between $r \in [2, 4]$), on both benchmark and real-world networks. This assures that
the breadth-first search must rarely go further than 1–3 steps from a node, keeping the algorithm efficient. We also note that a breadth-first search starting from each node is not required. Once a node is identified to be a generator point, the other nodes in its r neighborhood no longer need to be considered as potential local maxima. Overall, the complexity of detecting generator nodes stays well below $O(Mg)$, where g is the number of communities. (iii) Third, we partition the graph into Voronoi cells, a process with complexity of $O(N \log N)$.

4. Results

4.1. Benchmark testing

As argued by Fortunato et al in [3], community detection algorithms need to be tested both on benchmark graphs (generated with ground-truth; i.e., predetermined communities) and real-world networks. There are methods designed to generate benchmark networks with different properties [28, 29], using the following parameters: number of vertices N, average degree $\langle k \rangle$, mixing parameter $\mu$ (which means that each node shares a fraction $1 - \mu$ of its links with the nodes of its own community and a fraction $\mu$ with the nodes of other communities), maximum degree $k_{max}$, exponent of the degree distribution $\gamma$, exponent of the community size distribution $\beta$, maximal size of communities $s_{max}$, and minimal size of communities $s_{min}$ (for details see the supporting information).

We tested our method on benchmark networks with a large variety of parameter settings. Figure 3 shows the results obtained with 12 of these settings. 100 graphs were considered for each case, with $N = 1000$, $k_{max} = 200$, $\gamma = -2$, $\beta = -1$, $s_{min} = 50$, $s_{max} = 500$ and varying average degrees $\langle k \rangle = 6, 10, 15, 20$ and mixing parameters $\mu = 0.1, 0.2, 0.3$. We performed several rounds of Voronoi partitioning on each graph with increasing $r$, and monitored quality functions that compare our partitioning to the predefined communities of the graph. We used the widely accepted modularity function [4] (red line) and the less frequent separability [31] (blue line). The modularity function is defined as $Q = \frac{1}{2m} \sum (A_{ij} - P_{ij}) \delta (V_i, V_j)$, where the sum runs over all pairs of vertices, $A$ is the adjacency matrix, $m$ the total number of edges of the graph, and $P_{ij}$ represents the expected number of edges between vertices $i$ and $j$ in the null model (in the present case, the random graph model). The $\delta$ function yields one if vertices $i$ and $j$ are in the same community ($V_i = V_j$), zero otherwise [4]. The cluster level separability, on the other hand, is simply the ratio between the internal and external number of edges of a cluster: $g = \frac{m}{c + m}$, where $m$ and $c$ are the internal and external number of edges, respectively. The network level separability of a partitioning is the cluster size weighted average of the cluster level separabilities [31]. Depending on the concrete application, other metrics may be used. As the clusters in benchmark networks are predefined, we also computed the mutual information between our clustering and the original ground-truth (GT) cluster membership [30] (green line). The horizontal line in figure 3 shows the modularity of the original clustering of the benchmark graph (GT modularity). Our method reaches this maximum level of modularity as the local radius increases. At this optimal value of $r$, the mutual information is typically close to one, indicating that our method identifies the original clusters almost perfectly. Separability also increases with $r$, reaching a plateau at the same $r$ as modularity. As this quality function
saturates at large $r$ (occasionally showing a sudden increase), its usefulness as an indicator of optimal clustering is questionable. This phenomenon follows from the definition of separability, which naturally increases at small cluster numbers (it reaches its maximal value 1 at a single cluster).

Our algorithm achieves modularity values that are very close to their GT value, even in very sparse graphs with large mixing ($\langle k \rangle = 6, \mu = 0.3$). The community structure in these graphs is not very clear (indicated by the smaller value of the GT modularity); there is no good ground-truth to detect. Indeed, on these graphs the mutual information between our partitioning and the predefined cluster membership is small, indicating that the GT-level
modularity can be achieved even with clusters that are very different from the original benchmark communities.

4.2. Real-world networks

Due to the method used to generate benchmark networks, they have well-defined structures with clearly separated communities (especially if the mixing parameter is relatively small). The structure of real-world networks is less obvious; detecting communities poses a greater challenge. Moreover, the performance of algorithms is harder to test, as the ‘ground-truth’ community structure is not known. Consequently, new techniques are tested by comparing them to already accepted methods. Here we compare the clustering results of different algorithms through goodness (quality) metrics. Specifically, we have chosen modularity, proven to be one of the most reliable quality indicators of community detection.

We tested our algorithm on several real-world networks frequently used in the literature, widely different in structure and origin: (1) The updated version of the collaboration network of scientists on the condensed matter archive at www.arxiv.org (CONDMAT). This network is based on preprints posted to the archive between 1 January 1995 and 31 March 2005 [22], and it contains 39 576 nodes and 175 692 links. (2) Protein interaction network for yeast (YEAST) [23] including 1845 nodes and 4405 edges. (3) Network of links between American political blogs (POLBLOGS) [24] with 1223 nodes and 19 087 edges. (4) Neural network of the nematode C. elegans (CEL) [25, 26] containing 297 nodes and 2359 links. (5) The famous network of friendships between the 34 members of the Zachary karate club linked with 78 edges [27].

First, we followed the evolution of the modularity function with increasing radius $r$ on each of the five networks. Figure 4(A) shows the modularity of the POLBLOGS (see figure S2 in the supporting information for the other networks). Modularity reaches a maximum or a plateau at very small $r \in [2, 4]$ for each of the five networks. In some cases, such as the C. elegans neural network, modularity reaches a clear maximum at an optimal $r$, then starts to decrease (figure S2 in the supporting information). In other cases, such as CONDMAT and YEAST, modularity converges to a maximum at low $r$, then stays at this plateau (see the supporting information for details). The most intriguing behavior is seen in POLBLOGS (figure 4(A)), where modularity has several local maxima. Three distinct peaks appear at very different $r$ values ($r = 2.276, 3.2, 12.0$), pointing to a hierarchical structure. Figures 4(B)–(D) show the size distribution of partitions obtained at these $r$ values (indicated by colored dots in figure 4(A)). In American politics we expect two large, crisp communities corresponding to the democratic (left) and republican (right) wings. Indeed, the force-layout algorithm separates the graph in two clusters with almost equal size (figures 4(E)–(G)). Our algorithm agrees, but it discovers additional detail. For example, at $r = 2.276$ we have three large communities. One of these is almost twice as large as either of the other two, and it corresponds to the democratic left wing (blue cluster in figure 4(E)). The right wing, on the other hand, is cut into two distinct clusters (orange and red). At $r = 3.2$, our partitioning identifies the two, almost equal political party-based communities showcased by the force-layout algorithm (figure 4(F)). At $r = 12.0$, however, we find a third and much smaller modularity peak with the corresponding clustering shown in figure 4(G). This suggests that a very cohesive but small moderate cluster was merged with the right-wing community. Owing to its high local density, it became the Voronoi seed for
this larger-scale right-wing cluster. It appears that about 20% of the left-wing blogs are closer to this centrist group than to their own community, resulting in their split.

Next, we compared our method with five widely used algorithms (for details, see the supporting information): (1) the Louvain method based on local modularity optimization [7]; (2) the label propagation algorithm (LPA) [12]; (3) GANXiS, also called SLPA for speaker-listener label propagation algorithm [32, 33]; (4) link-community detection (LCD) [9] and (5) Infomap (IM) [34, 35]. Our goal was to use efficient algorithms which can be tested on large networks with thousands of nodes. Figure 5 shows a quality comparison between different methods, applied on the five real-world networks. Figure 5(A) shows modularity of the communities detected by each algorithm in ascending order, sorted by network type. Figure 5(B) shows the modularity of individual methods relative to the average modularity across the different methods, sorted by method. Our algorithm provides the second best

![Figure 4. Testing on real-world networks. (A) Modularity of Voronoi partitioning as a function of r in the POLBLOGS network. At the three different values indicated by colored dots the size distribution of partitions is shown in (B) for r = 2.276, (C) r = 3.2 and (D) r = 12.0. (E),(F),(G). The Gephi graph drawing software represents the network using the ForceAtlas2 layout algorithm [21]. Colors of nodes indicate the clusters detected with our algorithm at (E) r = 2.276, (F) r = 3.2 and (G) r = 12.0.](image-url)
performance; the modularity it achieves is often close to maximum and always above average. We consider its reliability to be its most important feature. In contrast to our method, three other algorithms drastically fail on at least one network: the IM fails on POLBLOGS, while GANXiS and LPA fail on CEL. LCD, on the other hand, is always around the average, but its performance is clearly weaker than that of our Voronoi partitioning method. The only algorithm which performs better is Louvain. This is expected, as the Louvain algorithm is based on direct optimization of modularity, the very quality function we use to judge best performance.

5. Discussion

Voronoi diagrams are widely used for partitioning metric spaces. Their clear and simple definition is easily generalized to graphs. Here we showed that graph Voronoi diagrams are well suited for community detection on complex networks. We surmounted two major challenges: First, we defined a suitable distance measure between nodes. To this end, we chose the inverse of the edge clustering coefficient, $1/ECC$. ECC is frequently used to characterize the probability that two connected nodes belong to different clusters. Second, we provided a method to identify the Voronoi seeds or generator nodes that mark community centers. To this end we leveraged another simple measure, the relative local density. Nodes were chosen as Voronoi seeds if their relative local density was maximal in their neighborhood of radius $r$. Using these two measures, we showed that our method outperforms all but one of five commonly used methods capable of
partitioning large networks, only falling behind a method which directly maximizes the quality function used for our comparisons.

Distance measures on graphs can be defined in a variety of ways, all of which aim to quantify the probability that two nodes belong to the same community (high probability reflected by a small distance; for an extensive review see [3]). Some of these measures are computed directly from the network structure, such as the number of edge- (or vertex-) independent paths between vertices [37] or structural equivalence [36]. Other methods define distance based on random walks on networks, such as the commute-time between vertices [38] or the time of average first passage [11, 39]. Our choice of the ECC [10] as the distance measure was motivated by its simplicity and computational efficiency. It is possible that employing more complex measures will further improve the mapping between Voronoi partitions and network communities. We expect the trade-off between accuracy and efficiency to yield a whole spectrum of future Voronoi-type methods for different application domains.

The most important parameter of our method is the radius \( r \), which defines the range in which local density maxima are chosen as seeds. As the same \( r \) is used for each seed, we need to address whether the method can detect clusters with strongly varying sizes. The distribution of cluster sizes is a power-law in many real-world networks. Thus, large clusters are often accompanied by many small communities. Can our algorithm deal with these types of networks? Our choice of the edge clustering coefficient as the base of our distance measure guarantees that it can. When one imagines a complex network in space, it is natural to think that large clusters (with many nodes) ‘occupy more space’. Using our distance measure, however, this is not so in the metric space we define. As ECC values tend to be much larger inside large, very dense clusters, they are compressed in metric space and their diameter is not typically larger than that of small clusters (sometimes quite the opposite). Moreover, distances between separate clusters, i.e., the length of edges connecting different clusters, do not cover multiple orders of magnitude. Rather, cluster placement in our metric space is relatively homogeneous, even if the number of nodes in different clusters shows strong variations. This assures that constructing Voronoi cells can detect small communities next to large ones. This is in fact illustrated by the size distribution of clusters detected in CONDMAT and YEAST (figure S2 in the supporting information).

We propose that the best strategy for using our method is to perform Voronoi partitioning at increasing values of \( r \). This procedure can reveal interesting details in the community structure of networks at different hierarchical levels. When a network has no specific hierarchical structure, quality functions such as modularity reach a large value at small \( r \), after which the partitioning changes very slowly. The method is thus not sensitive to the parameter \( r \) and allows us to identify the correct number of communities whenever there is no further structure hidden within. The presence of such a structure or hierarchy is signaled by the fact that the quality function has several local maxima (e.g., POLBLOGS). Between these maxima lie critical \( r \) values at which the detected community structure undergoes drastic changes (see figure S2 in the supporting information). These changes often correspond to community mergers into higher-scale clusters. The methods for automated detection of a network’s structural hierarchy based on these mergers needs to be addressed in future research. Moreover, the relationship between hierarchies detected in this way and those found by other hierarchical community detection methods (e.g., multi-resolution methods [41] and hierarchical methods [42]) also needs to be addressed.
Here we focused on graph partitioning problems where nodes are assigned a strict community membership. An important future goal of our research is to extend our Voronoi diagram-based method to overlapping communities. We envision two approaches to tackle this. First, we envision a fuzzy clustering method [3] that uses the distance between individual nodes and Voronoi seeds as the measure of their membership in each community. Nodes that fall close to Voronoi cell borders are prime candidates for regions of overlap. Second, tracking changes in single-node community membership with small perturbations of the radius $r$ (in the radius value range where the overall community structure is largely unchanged) may also serve to pinpoint network regions where communities overlap.

A few well-known hierarchical and spectral methods of data clustering have already been adapted to graphs [17, 18], albeit often via mapping the graph itself onto Euclidean space. Adapting the large set of centroid- and density-based methods available in data mining could further enrich community detection on complex networks, potentially customizing them for concrete applications. Our graph Voronoi diagram-based method combines some key features of these algorithms, and we expect it to be further extendable. For example, centroid-based methods such as $k$-means clustering concentrate on finding optimal cluster centers (Voronoi cell seeds in Euclidean space) in order to minimize the overall distance of all points from these seeds [43]. Our current approach, on the other hand, selects Voronoi seeds based on node density. We expect that methods to further optimize the position of seed nodes would be promising for improving community detection. Along different lines, our distance measure opens the possibility of defining the concept of density reachability on complex networks, a core concept of density-based clustering methods. (Two points in Euclidean space are density-reachable if they are closer than a threshold distance, and the density in their neighborhood is large enough; for a precise description see [44, 45].) In summary, our method shows that an expediently chosen topology-based metric on graphs can open new ways to approach graph community detection, narrowing the gap between the relatively young field of community detection and the well-explored realm of cluster analysis in data mining.

Acknowledgments

We thank Zoltán Néda, Razvan Florian and Zsuzsa Sárközi for a critical reading of the manuscript and our partner Epistemio (www.epistemio.com) for support. This work was supported by the Romanian National Authority for Scientific Research (CNDI-UEFISCDI) grant PN-II-PT-PCCA-2011-3.2-0895.

References

[1] Newman M E J 2010 Networks: An Introduction (New York: Oxford University Press)
[2] Barabasi A L 2012 Network Science (www.barabasilab.neu.edu/networksciencebook)
[3] Fortunato S 2010 Community detection in graphs Phys. Rep. 486 75–174
[4] Newman M E J 2006 Modularity and community structure in networks Proc. Natl Acad. Sci. 103 8577–82
[5] Newman M E J and Girvan M 2004 Finding and evaluating community structure in networks Phys. Rev. E 69 026113
[6] Reichardt J and Bornholdt S 2006 Statistical mechanics of community detection Phys. Rev. E 74 016110
[7] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 Fast unfolding of communities in large networks J. Stat. Mech.: Theor. Exp. 10 P10008
[8] Girvan M and Newman M E J 2002 Community structure in social and biological networks Proc. Natl Acad. Sci. USA 99 7821–6
[9] Ahn Y Y, Bagrow J P and Lehmann S 2010 Link communities reveal multiscale complexity in networks Nature 466 761–4
[10] Radicchi F, Castellano C, Cecconi F, Loreto V and Parisi D 2004 Defining and identifying communities in networks Proc. Natl Acad. Sci. USA 101 2658–63
[11] Zhou H 2003 Distance, dissimilarity index, and network community structure Phys. Rev. E 67 061901
[12] Ahn Y Y, Bagrow J P and Lehmann S 2010 Link communities reveal multiscale complexity in networks Nature 466 761–4
[13] Theodoridis S and Koutroumbas K 2008 Pattern Recognition (San Diego, CA: Academic Press)
[14] Han J, Kamber M and Pei J 2011 Data Mining: Concepts and Techniques (Burlington, MA: Morgan Kaufmann)
[15] Aurenhammer F 1991 Voronoi diagrams—a survey of a fundamental geometric data structure ACM Comput. Surv. 23 345–405
[16] Jain A K, Murty M N and Flynn P J 1999 Data clustering: a review ACM Comput. Surv. 31 264–323
[17] Nishikawa T and Motter A E 2011 Discovering network structure beyond communities Sci. Rep. 1 151
[18] Jin H, Wang S and Lib C 2013 Community detection in complex networks by density-based clustering Physica A 392 4606–18
[19] Chartrand G and Zhang P 2012 A First Course in Graph Theory (New York: Dover)
[20] Erwig M 2000 The graph Voronoi diagram with applications Networks 36 156–63
[21] Bastian M, Heymann S and Jacomy M 2009 Gephi: an open source software for exploring and manipulating networks ICWSM: AAAI Int. Conf. on Weblogs and Social Media pp 361–2
[22] Newman M E J 2001 The structure of scientific collaboration networks Proc. Natl Acad. Sci. USA 98 404–9
[23] Jeong H, Mason S, Barabási A L and Oltvai Z N 2001 Lethality and centrality in protein networks Nature 411 41–2
[24] Adamic L A and Glance N 2005 The political blogosphere and the 2004 US election: divided they blog Proc. 3rd Int. Workshop on Link Discovery (New York: ACM) pp 36–43
[25] White J G, Southgate E, Thompson J N and Brenner S 1986 The structure of the nervous system of the nematode Caenorhabditis elegans Phil. Trans. R. Soc. London 314 1–340
[26] Watts D J and Strogatz S H 1998 Collective dynamics of ‘small-world’ networks Nature 393 440–2
[27] Zachary W W 1977 An information flow model for conflict and fission in small groups J. Anthropol. Res. 33 452–73
[28] Lancichinetti A, Fortunato S and Radicchi F 2008 Benchmarks graphs for testing community detection algorithms Phys. Rev. E 78 046110
[29] Lancichinetti A and Fortunato S 2009 Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities Phys. Rev. E 80 016118
[30] Danon L, Diaz-Guilera A, Duch J and Arenas A 2005 Comparing community structure identification J. Stat. Mech.: Theor. Exp. P09008
[31] Yang J and Leskovec J 2012 Defining and evaluating network communities based on ground-truth ICDM2012: 12th IEEE Int. Conf. on Data Mining pp 745–54
[32] Xie J and Szymanski B 2012 Towards linear time overlapping community detection in social networks Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science vol 7302) (Berlin: Springer) pp 25–36
[33] Xie J, Szymanski B and Liu X 2011 SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process IEEE ICDM Workshop on DMCCI pp 344–9
[34] Rosvall M and Bergstrom C T 2008 Maps of random walks on complex networks reveal community structure Proc. Natl Acad. Sci. 105 1118–23
[35] Rosvall M, Axelsson D and Bergstrom C T 2009 The map equation Eur. Phys. J. Spec. Top. 178 13–23
[36] Lorraine F and White H J 1971 Structural equivalence of individuals in social networks J. Math. Sociol. 1 49–80
[37] Ahuja R K, Magnanti T L and Orlin J B 1993 Network Flows: Theory, Algorithms, and Applications (Englewood Cliffs, NJ: Prentice Hall)
[38] Fouss F, Renders J M, Saerens M and Pirotte A 2007 Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation IEEE. Trans. Knowl. Data Eng. 19 355–69
[39] White S and Smyth P 2005 A spectral clustering approach to finding communities in graphs Proc. of SIAM Int. Conf. on Data Mining pp 274–85
[40] Csardi G and Nepusz T 2006 The igraph software package for complex network research InterJournal Complex Systems 1695
[41] Arenas A, Fernández A and Gómen S 2008 Analysis of the structure of complex networks at different resolution level New J. Phys. 9 176
[42] Sales-Pardo M, Guimera R, Moreira A A and Amaral L A N 2007 Extracting the hierarchical organisation of complex systems Proc. Natl Acad. Sci. 104 15224
[43] Lloyd S 1982 Least squares quantization in PCM IEEE Trans. Inform. Theory 28 129–37
[44] Ester M, Kriegel H P, Sander J and Xiaowei X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise KDD ’96: Proc. 2nd Int. Conf. on Discovery and Data Mining (Menlo Park, CA: AAAI) pp 226–31
[45] Ankerst M, Breunig M, Kriegel H P and Sander J 1999 OPTICS: ordering points to identify the clustering structure ACM SIGMOD Record 28 49–60