

# Local Network Community Detection with Continuous Optimization of Conductance and Weighted Kernel K-Means

**Twan van Laarhoven**

TVANLAARHOVEN@CS.RU.NL

**Elena Marchiori**

ELENAM@CS.RU.NL

*Institute for Computing and Information Sciences*

*Radboud University Nijmegen*

*Postbus 9010*

*6500 GL Nijmegen, The Netherlands*

**Editor:** Jure Leskovec

## Abstract

Local network community detection is the task of finding a single community of nodes concentrated around few given seed nodes in a localized way. Conductance is a popular objective function used in many algorithms for local community detection. This paper studies a continuous relaxation of conductance. We show that continuous optimization of this objective still leads to discrete communities. We investigate the relation of conductance with weighted kernel k-means for a single community, which leads to the introduction of a new objective function,  $\sigma$ -conductance. Conductance is obtained by setting  $\sigma$  to 0. Two algorithms, EMC and PGDC, are proposed to locally optimize  $\sigma$ -conductance and automatically tune the parameter  $\sigma$ . They are based on expectation maximization and projected gradient descent, respectively. We prove locality and give performance guarantees for EMC and PGDC for a class of dense and well separated communities centered around the seeds. Experiments are conducted on networks with ground-truth communities, comparing to state-of-the-art graph diffusion algorithms for conductance optimization. On large graphs, results indicate that EMC and PGDC stay localized and produce communities most similar to the ground, while graph diffusion algorithms generate large communities of lower quality.<sup>1</sup>

**Keywords:** community detection, stochastic block model, conductance

## 1. Introduction

Imagine that you are trying to find a community of nodes in a network around a given set of nodes. A simple way to approach this problem is to consider this set as seed nodes, and then keep adding nodes in a local neighborhood of the seeds as long as this makes the community better in some sense. In contrast to global clustering, where the overall community structure of a network has to be found, local community detection aims to find only one community around the given seeds by relying on local computations involving only nodes relatively close to the seed. Local community detection by seed expansion is

---

1. Source code of the algorithms used in the paper is available at <http://cs.ru.nl/~tvanlaarhoven/conductance2016>.

especially beneficial in large networks, and is commonly used in real-life large scale network analysis (Gargi et al., 2011; Leskovec et al., 2010; Wu et al., 2012).

Several algorithms for local community detection operate by seed expansion. These methods have different expansion strategies, but what they have in common is their use of conductance as the objective to be optimized. Intuitively, conductance measures how strongly a set of nodes is connected to the rest of the graph; sets of nodes that are isolated from the graph have low conductance and make good communities.

The problem of finding a set of minimum conductance in a graph is computationally intractable (Chawla et al., 2005; Šíma and Schaeffer, 2006). As a consequence, many heuristic and approximation algorithms for local community detection have been introduced (see references in the related work section). In particular, effective algorithms for this task are based on the local graph diffusion method. A graph diffusion vector  $\mathbf{f}$  is an infinite series  $\mathbf{f} = \sum_{i=0}^{\infty} \alpha_i \mathbf{P}^i \mathbf{s}$ , with diffusion coefficients  $\sum_{i=0}^{\infty} \alpha_i = 1$ , seed nodes  $\mathbf{s}$ , and random walk transition matrix  $\mathbf{P}$ . Types of graph diffusion, such as personalized Page Rank (Andersen and Lang, 2006) and Heat Kernel (Chung, 2007), are determined by the choice of the diffusion coefficients. In the diffusion method an approximation of  $\mathbf{f}$  is computed. After dividing each vector component by the degree of the corresponding node, the nodes are sorted in descending order by their values in this vector. Next, the conductance of each prefix of the sorted list is computed and either the set of smallest conductance is selected, e.g. in (Andersen and Lang, 2006) or a local optima of conductance along the prefix length dimension (Yang and Leskovec, 2012) is considered.

These algorithms optimize conductance along a single dimension, representing the order in which nodes are added by the algorithm. However this ordering is mainly related to the seed, and not directly to the objective that is being optimized. Algorithms for the direct optimization of conductance mainly operate in the discrete search space of communities, and locally optimize conductance by adding and/or removing one node. This amounts to fixing a specific neighborhood structure over communities where the neighbors of a community are only those communities which differ by the membership of a single node. This is just one possible choice of community neighbor. A natural way to avoid the problem of choosing a specific neighborhood structure is to use continuous rather than discrete optimization. To do this, we need a continuous relaxation of conductance, extending the notion of communities to allow for fractional membership. This paper investigates such a continuous relaxation, which leads to the following findings.

### 1.0.1 ON LOCAL OPTIMA

Although local optima of a continuous relaxation of conductance might at first glance have nodes with fractional memberships, somewhat surprisingly all strict local optima are discrete. This means that continuous optimization can directly be used to find communities without fractional memberships.

### 1.0.2 RELATION WITH WEIGHTED KERNEL K-MEANS

We unravel the relation between conductance and weighted kernel k-means objectives using the framework by Dhillon et al. (2007). Since the aim is to find only one community, we consider a slight variation with one mean, that is, with  $k = 1$ . This relation leads

to the introduction of a new objective function for local community detection, called  $\sigma$ -conductance, which is the sum of conductance and a regularization term whose influence is controlled by a parameter  $\sigma$ . Interestingly, the choice of  $\sigma$  has a direct effect on the number of local optima of the function, where larger values of  $\sigma$  lead to more local optima. In particular, we prove that for  $\sigma > 2$  all discrete communities are local optima. As a consequence, due to the seed expansion approach, local optimization of  $\sigma$ -conductance favors smaller communities for larger values of  $\sigma$ .

### 1.0.3 ALGORITHMS

Local optimization of  $\sigma$ -conductance can be easily performed using the projected gradient descent method. We develop an algorithm based on this method, called PGDC. Motivated by the relation between conductance and k-means clustering, we introduce an Expectation-Maximization (EM) algorithm for  $\sigma$ -conductance optimization, called EMC. We show that for  $\sigma = 0$ , this algorithm is almost identical to projected gradient descent with an infinite step size in each iteration. We then propose a heuristic procedure for choosing  $\sigma$  automatically in these algorithms.

### 1.0.4 RETRIEVING COMMUNITIES

We give a theoretic characterization of a class of communities, called dense and isolated communities, for which PGDC and EMC perform optimally. For this class of communities the algorithms exactly recover a community from the seeds. We investigate the relation between this class of communities and the notion of  $(\alpha, \beta)$ -cluster proposed by (Mishra et al., 2008) for social networks analysis. And we show that, while all maximal cliques in a graph are  $(\alpha, \beta)$ -clusters, they are not necessarily dense and isolated communities. We give a simple condition on the degree of the nodes of a community which guarantees that a dense and isolated community satisfying such condition is also an  $(\alpha, \beta)$ -cluster.

### 1.0.5 EXPERIMENTAL PERFORMANCE

We use publicly available artificial and real-life network data with labeled ground-truth communities to assess the performance of PGDC and EMC. Results of the two methods are very similar, with PGDC performing slightly better, while EMC is slightly faster. These results are compared with those obtained by three state-of-the-art algorithms for conductance optimization based on the local graph diffusion: the popular Personalized Page Rank (PPR) diffusion algorithm by Andersen and Lang (2006), a more recent variant by Yang and Leskovec (2012) (here called YL), and the Heat Kernel (HK) diffusion algorithm by Kloster and Gleich (2014). On large networks PGDC and EMC stay localized and produce communities which are more faithful to the ground truth than those generated by the considered graph diffusion algorithms. PPR and HK produce much larger communities with a low conductance, while the YL strategy outputs very small communities with a higher conductance.

## 1.1 Related Work

The enormous growth of network data from diverse disciplines such as social and information science and biology has boosted research on network community detection (see for instance the overviews by Schaeffer (2007) and Fortunato (2010)). Here we confine ourselves to literature we consider to be relevant to the present work, namely local community detection by seed expansion, and review related work on conductance as objective function and its local optimization. We also briefly review research on other objective functions, and on properties of communities and of seeds.

### 1.1.1 CONDUCTANCE AND ITS LOCAL OPTIMIZATION

Conductance has been largely used for network community detection. For instance Leskovec et al. (2008) introduced the notion of network community profile plot to measure the quality of a ‘best’ community as a function of community size in a network. They used conductance to measure the quality of a community and analyze a large number of communities of different size scales in real-world social and information networks.

Direct conductance optimization was shown to favor communities which are quasi-cliques (Kang and Faloutsos, 2011) or communities of large size which include irrelevant subgraphs (Andersen and Lang, 2006; Whang et al., 2013).

Popular algorithms for local community detection employ the local graph diffusion method to find a community with small conductance.

Starting from the seminal work by Spielman and Teng (2004) various algorithms for local community detection by seed expansion based on this approach have been proposed (Andersen et al., 2006; Avron and Horesh, 2015; Chung, 2007; Kloster and Gleich, 2014; Zhu et al., 2013a). The theoretical analysis in these works is largely based on a mixing result which shows that a cut with small conductance can be found by simulating a random walk starting from a single node for sufficiently many steps (Lovász and Simonovits, 1990). This result is used to prove that if the seed is near to a set with small conductance then the result of the procedure is a community with a related conductance, which is returned in time proportional to the volume of the community (up to a logarithmic factor).

Mahoney et al. (2012) performed local community detection by modifying the spectral program used in standard global spectral clustering. Specifically the authors incorporated a bias towards a target region of seed nodes in the form of a constraint to force the solution to be well connected with or to lie near the seeds. The degree of connectedness was specified by setting a so-called correlation parameter. The authors showed that the optimal solution of the resulting constrained optimization problem is a generalization of Personalized PageRank (Andersen and Lang, 2006).

### 1.1.2 OTHER OBJECTIVES

Conductance is not the only objective function used in local community detection algorithms. Various other objective functions have been considered in the literature. For instance, Chen et al. (2009) proposed to use the ratio of the average internal and external degree of nodes in a community as objective function. Clauset (2005) proposed a local variant of modularity. Wu et al. (2015) modified the classical density objective, equal to the sum of edges in the community divided by its size, by replacing the denominator with the

sum of weights of the community nodes, where the weight of a node quantifies its proximity to the seeds and is computed using a graph diffusion method.

A comparative experimental analysis of objective functions with respect to their experimental and theoretical properties was performed e.g. in (Yang and Leskovec, 2012) and (Wu et al., 2015), respectively.

### 1.1.3 PROPERTIES OF COMMUNITIES

Instead of focusing on objective functions and methods for local community detection, other researchers investigated properties of communities. Mishra et al. (2008) focused on interesting classes of communities and algorithms for their exact retrieval. They defined the so called  $(\alpha, \beta)$ -communities and developed algorithms capable of retrieving this type of communities starting from a seed connected to a large fraction of the members of the community. Zhu et al. (2013b) considered the class of well-connected communities, which have a better internal connectivity than conductance. Internal connectivity of a community is defined as the inverse of the mixing time for a random walk on the subgraph induced by the community. They showed that for well-connected communities, it is possible to provide an improved performance guarantee, in terms of conductance of the output, for local community detection algorithms based on the diffusion method. Gleich and Seshadhri (2012) investigated the utility of neighbors of the seed; in particular they showed empirically that such neighbors form a ‘good’ local community around the seed. Yang and Leskovec (2012) investigated properties of ground truth communities in social, information and technological networks.

Lancichinetti et al. (2011) addressed the problem of finding a significant local community from an initial group of nodes. They proposed a method which locally optimizes the statistical significance of a community, defined with respect to a global null model, by iteratively adding external significant nodes and removing internal nodes that are not statistically relevant. The resulting community is not guaranteed to contain the nodes of the initial community.

### 1.1.4 PROPERTIES OF SEEDS

Properties of seeds in relation to the performance of algorithms were investigated by e.g. Kloumann and Kleinberg (2014). They considered different types of algorithms, in particular a greedy seed expansion algorithm which at each step adds the node that yields the most negative change in conductance (Mislove et al., 2010). Whang et al. (2013) investigated various methods for choosing the seeds for a PageRank based algorithm for community detection. Chen et al. (2013) introduced the notion of local degree central node, whose degree is greater than or equal to the degree of its neighbor nodes. A new local community detection method is introduced based on the local degree central node. In this method, the local community is not discovered from the given starting node, but from the local degree central node that is associated with the given starting node.

## 1.2 Notation

We start by introducing the notation used in the rest of this paper. We denote by  $V$  the set of nodes in a network or graph  $G$ . A community, also called a cluster,  $C \subseteq V$  will be a

subset of nodes, and its complement  $\bar{C} = V \setminus C$  consists of all nodes not in  $C$ . Note that we consider any subset of nodes to be a community, and the goal of community detection is to find a *good* community.

Let  $A$  be the adjacency matrix of  $G$ , where  $a_{ij}$  denotes the weight of an edge between nodes  $i$  and  $j$ . In unweighted graphs  $a_{ij}$  is either 0 or 1, and in undirected graphs  $a_{ij} = a_{ji}$ . In this paper we work only with unweighted undirected graphs. We can generalize this notation to sets of nodes, and write  $a_{xy} = \sum_{i \in x} \sum_{j \in y} a_{ij}$ . With this notation in hand we can write conductance as

$$\phi(C) = \frac{a_{C\bar{C}}}{a_{CV}} = 1 - \frac{a_{CC}}{a_{CV}}.$$

A common alternative definition is

$$\phi_{\text{alt}}(C) = \frac{a_{C\bar{C}}}{\min(a_{CV}, a_{\bar{C}V})},$$

which considers the community to be the smallest of  $C$  and  $\bar{C}$ . For instance Kloster and Gleich (2014) and Andersen and Lang (2006) use this alternative definition, while Yang and Leskovec (2012) use  $\phi$ .

Note that  $\phi$  has a trivial optimum when all nodes belong to the community, while  $\phi_{\text{alt}}$  will usually have a global optimum with roughly half of the nodes belonging to the community. Neither of these optima are desirable for finding a single small community.

With a set  $X$  we associate an indicator vector  $[X]$  of length  $|V|$ , such that  $[X]_i = 1$  if  $i \in X$  and  $[X]_i = 0$  otherwise. We will usually call this vector  $\mathbf{x}$ .

## 2. Continuous Relaxation of Conductance

If we want to talk about directly optimizing conductance, then we need to define what (local) optima are. The notion of local optima depends on the topology of the input space, that is to say, on what communities we consider to be neighbors of other communities. We could, for instance, define the neighbors of a community to be all communities that can be created by adding or removing a single node. But this is an arbitrary choice, and we could equally well define the neighbors to be all communities reached by adding or removing up to two nodes. An alternative is to move to the continuous world, where we can use our knowledge of calculus to give us a notion of local optima.

To turn community finding into a continuous problem, instead of a set  $C$  we need to see the community as a vector  $\mathbf{c}$  of real numbers between 0 and 1, where  $c_i$  denotes the degree to which node  $i$  is a member of the community. Given a discrete community  $C$ , we have  $\mathbf{c} = [C]$ , but the inverse is not always possible, so the vectorial setting is more general.

The edge weight between sets of nodes can be easily generalized to the edge weight of membership vectors,

$$a_{\mathbf{x}\mathbf{y}} = \mathbf{x}^T \mathbf{A} \mathbf{y} = \sum_{i \in V} \sum_{j \in V} x_i a_{ij} y_j.$$

Now we can reinterpret the previous definition of conductance as a function of real vectors, which we could expand as

$$\phi(\mathbf{c}) = 1 - \frac{\sum_{i,j \in V} c_i a_{ij} c_j}{\sum_{i,j \in V} c_i a_{ij}}.$$

With this definition we can apply the vast literature on constrained optimization of differentiable functions. In particular, we can look for local optima of the conductance, subject to the constraint that  $0 \leq c_i \leq 1$ . These local optima will satisfy the Karush-Kuhn-Tucker conditions, which in this case amounts to, for all  $i \in V$ ,

$$\begin{aligned} 0 &\leq c_i \leq 1 \\ \nabla\phi(\mathbf{c})_i &\geq 0 \quad \text{if } c_i = 0 \\ \nabla\phi(\mathbf{c})_i &= 0 \quad \text{if } 0 < c_i < 1, \\ \nabla\phi(\mathbf{c})_i &\leq 0 \quad \text{if } c_i = 1. \end{aligned}$$

To use the above optimization problem for finding communities from seeds, we add one additional constraint. Given a set  $S$  of seeds we require that  $c_i \geq s_i$ ; in other words, that the seed nodes are members of the community. This is the only way in which the seeds are used, and the only way in which we can use the seeds without making extra assumptions.

### 2.1 A Look at the Local Optima

By allowing community memberships that are real numbers, uncountably many more communities are possible. One might expect that it is overwhelmingly likely that optima of the continuous relaxation of conductance are communities with fractional memberships. But this turns out not to be the case. In fact, the strict local optima will all represent discrete communities.

To see why this is the case, consider the objective in terms of the membership coefficient  $c_i$  for some node  $i$ . This takes the form of a quadratic rational function,

$$\phi(c_i) = \frac{\alpha_1 + \alpha_2 c_i + \alpha_3 c_i^2}{\alpha_4 + \alpha_5 c_i}.$$

The coefficients in the denominator are positive, which means that the denominator is also positive for  $c_i > 0$ . At an interior local minimum we must have  $\phi'(c_i) = 0$ , which implies that  $\phi''(c_i) = 2\alpha_3/(\alpha_4 + \alpha_5 c_i)^3$ . But  $\alpha_3 \leq 0$ , since it comes from the  $c_i a_{ii} c_i$  term in the numerator of the conductance, so  $\phi''(c_i) \leq 0$ , and hence there are only local maxima or saddle points, not strict local minima.

It is still possible for there to be plateaus in the objective functions, where  $\phi(\mathbf{c})$  is optimal regardless of the value of  $c_i$  for a certain node  $i$ .

### 2.2 The Relation to Weighted Kernel K-Means Clustering

Another view on conductance is by the connection to weighted kernel  $k$ -means clustering. The connection between weighted kernel  $k$ -means and objectives for graph partitioning has been thoroughly investigated in Dhillon et al. (2007). Here we extend that connection to the single cluster case.

Start with weighted  $k$ -means clustering, which, given a dataset  $\{x_i\}_{i=1}^N$  and weights  $\{w_i\}$ , minimizes the following objective

$$\sum_{i=1}^N \sum_{j=1}^k w_i c_{ij} \|x_i - \mu_j\|_2^2$$

with respect to  $\mu_j$  and  $c_{ij}$ , where  $c_{ij}$  indicates if point  $i$  belongs to cluster  $j$ , subject to the constraint that exactly one  $c_{ij}$  is 1 for every  $i$ .

Since our goal is to find a single cluster, a first guess would be to take  $k = 2$ , and to try to separate a foreground cluster from the background. But when using 2-means, there is no distinction between foreground and background, and so solutions will naturally have two clusters of roughly equal size. Instead, we can consider a one-cluster variant that distinguishes between points in a cluster and background points, which we call 1-mean clustering. This can be formulated as the minimization of

$$\sum_i w_i (c_i \|x_i - \mu\|_2^2 + (1 - c_i) \lambda_i)$$

with respect to a single  $\mu$  and cluster membership indicators  $c_i$  (between 0 and 1). Here  $\lambda_i$  is a cost for node  $i$  being a member of the background.

We allow different  $\lambda_i$  for different nodes, as there is no reason to demand a single value. The condition for a node  $i$  to be part of the community is  $\|x_i - \mu\|_2^2 < \lambda_i$ . So different values for  $\lambda_i$  might be useful for two reasons. The first would be to allow incorporating prior knowledge, the second reason would be if the scale (of the clusters) is different, that is, nodes (in different clusters) have different distances from the mean. By adding a diagonal matrix to the kernel, the squared distance from all points to all other points is increased by that same amount. It makes sense to compensate for this in the condition for community membership. And since the diagonal terms we add to the kernel vary per node, the amount that these nodes move away from other points also varies, which is why we use different  $\lambda_i$  per node.

The minimizer for  $\mu$  is the centroid of the points inside the cluster,

$$\mu = \frac{\sum_i w_i c_i x_i}{\sum_i w_i c_i};$$

while the minimizer for  $c_i$  is 1 if and only if  $\|x_i - \mu\|_2^2 < \lambda_i$ , and 0 otherwise.

The  $k$ -means and 1-mean objectives can be kernelized by writing distances in terms of inner products, and using a kernel  $K(i, j) = \langle x_i, x_j \rangle$ . The cluster mean is then a linear combination of points,  $\mu = \sum_i \mu_i x_i$ , giving

$$\|x_i - \mu\|_2^2 = K(i, i) - 2 \sum_j \mu_j K(i, j) + \sum_{j, k} \mu_j K(j, k) \mu_k.$$

By filling in the optimal  $\mu$  given above, the 1-mean objective then becomes

$$\begin{aligned} \phi_{W, K, \lambda}(\mathbf{c}) &= \sum_i w_i c_i (K(i, i) - \lambda_i) + \sum_i w_i \lambda_i \\ &\quad - \frac{\sum_{i, j} w_i c_i w_j c_j K(i, j)}{\sum_i w_i c_i}. \end{aligned}$$

The second term is constant, so we can drop it for the purposes of optimization.

We pick  $\lambda_i = K(i, i)$ . With this choice, the condition for a node  $i$  to be a member of the community is  $\|x_i - \mu\|_2^2 < \|x_i - 0\|_2^2$ . This can be seen as a 2-means cluster assignment



where the background cluster has the origin as the fixed mean. With this choice the first term also drops out.

By converting the graph into a kernel with

$$K = W^{-1}AW^{-1},$$

where  $W$  is a diagonal matrix with the weights  $w_i$  on the diagonal, we can obtain objectives like conductance and association ratio. However this  $K$  is not a legal kernel, because a kernel has to be positive definite. Without a positive definite kernel the distances  $\|x_i - \mu\|$  from the original optimization problem can become negative. To make the kernel positive definite, we follow the same route as Dhillon et al., and add a diagonal matrix, obtaining

$$K = \sigma W^{-1} + W^{-1}AW^{-1}.$$

Since we are interested in conductance, we take as weights  $w_i = a_{iV}$ , the degree of node  $i$ , and we take  $\lambda_i = K(i, i)$ . This results (up to an additive constant) in the following objective which we call  $\sigma$ -conductance,

$$\phi_\sigma(\mathbf{c}) = 1 - \frac{\sum_{i,j} c_i c_j a_{ij}}{\sum_i c_i a_{iV}} - \sigma \frac{\sum_i c_i^2 a_{iV}}{\sum_i c_i a_{iV}}.$$

Observe that if  $\mathbf{c}$  is a discrete community, then  $c_i^2 = c_i$ , and the last term is constant. In that case optimization of this objective is exactly equivalent to optimizing conductance.

For the purposes of continuous optimization however, increasing the  $\sigma$  parameter has the effect of increasing the objective value of non-discrete communities. So different communities become more separated, and in the extreme case, every discrete community becomes a local optimum.

**Theorem 1** *When  $\sigma > 2$ , all discrete communities  $\mathbf{c}$  are local minima of  $\phi_\sigma(\mathbf{c})$  constrained to  $0 \leq c_i \leq 1$ .*

**Proof** The gradient of  $\phi_\sigma$  is

$$\nabla \phi_\sigma(\mathbf{c})_i = a_{iV} \frac{a_{\mathbf{c}\mathbf{c}}}{a_{\mathbf{c}V}^2} - 2 \frac{a_{i\mathbf{c}}}{a_{\mathbf{c}V}} + \sigma \left( a_{iV} \frac{\sum_j c_j^2 a_{jV}}{a_{\mathbf{c}V}^2} - 2c_i \frac{a_{iV}}{a_{\mathbf{c}V}} \right).$$

When  $\mathbf{c}$  is discrete, then  $\sum_j c_j^2 a_{jV} = a_{\mathbf{c}V}$ , so the gradient simplifies to

$$\nabla \phi_\sigma(\mathbf{c})_i = \frac{a_{iV}}{a_{\mathbf{c}V}} \left( \frac{a_{\mathbf{c}\mathbf{c}}}{a_{\mathbf{c}V}} + (1 - 2c_i)\sigma - 2 \frac{a_{i\mathbf{c}}}{a_{iV}} \right).$$

Because  $a_{i\mathbf{c}} \leq a_{iV}$  and  $a_{\mathbf{c}\mathbf{c}} \leq a_{\mathbf{c}V}$  we can bound this by

$$\frac{a_{iV}}{a_{\mathbf{c}V}} ((1 - 2c_i)\sigma - 2) \leq \nabla \phi_\sigma(\mathbf{c})_i \leq \frac{a_{iV}}{a_{\mathbf{c}V}} ((1 - 2c_i)\sigma + 1).$$

So if  $c_i = 0$ , we get that  $\nabla \phi_\sigma(\mathbf{c})_i > 0$  when  $\sigma > 2$ . And if  $c_i = 1$ , we get that  $\nabla \phi_\sigma(\mathbf{c})_i < 0$  when  $\sigma > 1$ .

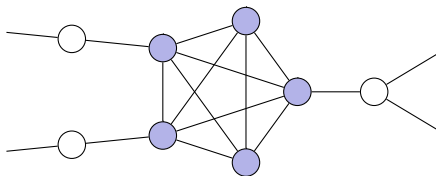


Figure 1: A simple subnetwork consisting of a clique with tails connecting it to the rest of the network. The clique (shaded nodes) is not a local optimum of conductance, but it is a local optimum of  $\sigma$ -conductance when  $\sigma > 0.131$ .

This means that when  $\sigma > 2$  all discrete communities satisfy the KKT conditions, and from the sign of the gradient we can see that they are not local maxima. Furthermore,  $\phi_\sigma(c)$  is a concave function, so it has no saddle points (see the proof of Theorem 2). This means that all discrete communities are local minima of  $\phi_\sigma$ . ■

Conversely, the result from Section 2.1 generalizes to  $\sigma$ -conductance,

**Theorem 2** *When  $\sigma \geq 0$ , all strict local minima  $\mathbf{c}$  of  $\phi_\sigma(\mathbf{c})$  constrained to  $0 \leq c_i \leq 1$  are discrete. Furthermore, if  $\sigma > 0$  then all local minima are discrete.*

**Proof** By the argument from Section 2.1. When  $\sigma > 0$  it is always the case that  $\alpha_3 < 0$ , so there are no saddle points or plateaus, and all local minima are discrete. ■

As an example application of  $\sigma$ -conductance, consider the network in Figure 1. In this network, the clique is not a local optimum of regular conductance. This is because the gradient for the adjacent nodes with degree 2 is always negative, regardless of the conductance of the community. However, for  $\sigma$ -conductance this gradient becomes positive when  $\sigma > \phi_\sigma(\mathbf{c})$ , in this case when  $\sigma > 0.131$ . In other words, with higher  $\sigma$ , adjacent nodes with low degree are no longer considered part of otherwise tightly connected communities such as cliques.

### 3. Algorithms

We now introduce two simple algorithms for the local optimization of conductance and  $\sigma$ -conductance, analyze their computational complexity and provide an exact performance guarantee for a class of communities. Then we look at a procedure for the automatic selection of a value for  $\sigma$ .

#### 3.1 Projected Gradient Descent

Perhaps the simplest possible method for constrained continuous optimization problems is projected gradient descent. This is an iterative algorithm, where in each step the solution is moved in the direction of the negative gradient, and then this solution is projected so as to satisfy the constraints.

In our case, we start from an initial community containing only the seeds,

$$\mathbf{c}^{(0)} = \mathbf{s},$$

where  $\mathbf{s} = [S]$  is a sparse vector indicating the seed node(s). Then in each subsequent iteration we get

$$\mathbf{c}^{(t+1)} = p(\mathbf{c}^{(t)} - \gamma^{(t)} \nabla \phi(\mathbf{c}^{(t)})).$$

This process is iterated until convergence. The step size  $\gamma^{(t)}$  can be found with line search. The gradient  $\nabla \phi$  is given by

$$\nabla \phi(\mathbf{c})_i = \frac{a_{iV} a_{\mathbf{c}\mathbf{c}}}{a_{\mathbf{c}V}^2} - 2 \frac{a_{i\mathbf{c}}}{a_{\mathbf{c}V}}.$$

And the projection  $p$  onto the set of valid communities is defined by

$$p(\mathbf{c}) = \underset{\mathbf{c}', \text{ s.t. } 0 \leq c'_i \leq 1, s_i \leq c'_i}{\operatorname{argmin}} \|\mathbf{c} - \mathbf{c}'\|_2^2,$$

which simply amounts to

$$p(\mathbf{c}) = \max(\mathbf{s}, \min(1, \mathbf{c})).$$

This function clips values above 1 to 1, and values below  $s_i$  to  $s_i$ . Since  $s_i \geq 0$  this also enforces that  $c_i \geq 0$ .

The complete algorithm is given in Algorithm 1. If a discrete community is desired, as a final step, we might threshold the vector  $\mathbf{c}$ . But as shown in Theorem 2 the found community is usually already discrete.

### 3.2 Expectation-Maximization

The connection to  $k$ -means clustering suggests that it might be possible to optimize conductance using an Expectation-Maximization algorithm similar to Lloyd's algorithm for  $k$ -means clustering. Intuitively, the algorithm would work as follows:

- **E step** assign each node  $i$  to the community if and only if its squared distance to the mean is less than  $\lambda_i$ .
- **M step** set the community mean to the weighted centroid of all nodes in the community.

These steps are alternated until convergence. Since both these steps do not increase the objective value, the algorithm is guaranteed to converge.

If the community after some iterations is  $C$ , then, as in the previous section, we can fill in the optimal mean into the E step, to obtain that a node  $i$  should be part of the community if

$$K(i, i) + \frac{a_{CC}/a_{CV} + \sigma}{a_{CV}} - 2 \frac{a_{iC}/a_{iV} + \sigma c_i}{a_{CV}} < \lambda_i.$$

When  $\lambda_i = K(i, i)$ , this condition is equivalent to

$$\nabla \phi_\sigma(C) < 0.$$

---

**Algorithm 1** Projected Gradient Descent conductance optimization (PGDC)

---

**Input:** A set  $S$  of seeds of seeds, a graph  $G$ , a constant  $\sigma \geq 0$ .

```

1:  $\mathbf{s} \leftarrow [S]$ 
2:  $\mathbf{c}^{(0)} \leftarrow \mathbf{s}$ 
3:  $t \leftarrow 0$ 
4: repeat
5:    $\gamma^{(t)} \leftarrow \text{LineSearch}(\mathbf{c}^{(t)})$ 
6:    $\mathbf{c}^{(t+1)} = p(\mathbf{c}^{(t)} - \gamma^{(t)} \nabla \phi_\sigma(\mathbf{c}^{(t)}))$ 
7:    $t \leftarrow t + 1$ 
8: until  $\mathbf{c}^{(t-1)} = \mathbf{c}^{(t)}$ 
9:  $C \leftarrow \{i \in V \mid \mathbf{c}_i^{(t)} \geq 1/2\}$ 

```

---

**function** LineSearch( $\mathbf{c}$ )

---

```

1:  $\gamma^* \leftarrow 0, \quad \phi^* \leftarrow \phi_\sigma(\mathbf{c})$ 
2:  $\mathbf{g} \leftarrow \nabla \phi_\sigma(\mathbf{c})$ 
3:  $\gamma \leftarrow 1/\max(|\mathbf{g}|)$ 
4: repeat
5:    $\mathbf{c}' \leftarrow p(\mathbf{c} - \gamma \mathbf{g})$ 
6:   if  $\phi_\sigma(\mathbf{c}') < \phi^*$  then
7:      $\gamma^* \leftarrow \gamma, \quad \phi^* \leftarrow \phi_\sigma(\mathbf{c}')$ 
8:   end if
9:    $\gamma \leftarrow 2\gamma$ 
10: until  $\mathbf{c}'_i \in \{0, 1\}$  for all  $i$  with  $g_i \neq 0$ 
11: return  $\gamma^*$ 

```

---



---

**Algorithm 2** EM conductance optimization (EMC)

---

**Input:** A set  $S$  of seeds, a graph  $G$ , a constant  $\sigma \geq 0$ .

```

1:  $C^{(0)} \leftarrow S$ 
2:  $t \leftarrow 0$ 
3: repeat
4:    $C^{(t+1)} = \{i \mid \nabla \phi_\sigma(C^{(t)})_i < 0\} \cup S$ 
5:    $t \leftarrow t + 1$ 
6: while  $C^{(t)} < C^{(t-1)}$ 

```

---

This leads us to the EM community finding algorithm, Algorithm 2.

By taking  $\sigma = 0$  we get that nodes are assigned to the community exactly if the gradient  $\nabla\phi(C)_i$  is negative. So, this EM algorithm is very similar to projected gradient descent with an infinite step size in each iteration. The only difference is for nodes with  $\nabla\phi(C)_i = 0$ , which in the EMC algorithm are always assigned to the background, while in PGD their membership of the community is left unchanged compared to the previous iteration.

Of course, we have previously established that  $\sigma = 0$  does not lead to a valid kernel (this doesn't preclude us from still using the EM algorithm). In the case that  $\sigma > 0$  there is an extra barrier for adding nodes not currently in the community, and an extra barrier for removing nodes that are in the community. This is similar to the effect that increasing  $\sigma$  has on the gradient of  $\phi_\sigma$ .

### 3.3 Computational Complexity

Both methods require the computation of the gradient in each iteration. This computation can be done efficiently. The only nodes for which the gradient of the conductance is negative are the neighbors of nodes in the current community, and the only nodes for which a positive gradient can have an effect are those in the community. So the gradient doesn't need to be computed for other nodes. For the other nodes the gradient depends on the number of edges to the community, and on the node's degree. Assuming that the node degree can be queried in constant time, the total time per iteration is proportional to the size of the one-step-neighborhood of the community, which is of the order of the volume of the community. If the node degrees are not known, then the complexity increases to be proportional to the volume of the one-step-neighborhood of the community, though this is a one-time cost, not a per iteration cost.

As seen in Section 3.5, for dense and isolated communities, the number of iterations is bounded by the diameter of the community. In general we can not guarantee such a bound, but in practice the number of iterations is always on the order of the diameter of the recovered community.

For very large datasets, the computation of the gradient can still be expensive, even though it is a local operation. Therefore, we restrict the search to a set of 1000 nodes near the seed. This set  $N$  is formed by starting with the seed, and repeatedly adding all neighbors of nodes in  $N$ , until the set would contain more than 1000 nodes. In this last step we only add the nodes with the highest  $a_{iN}/a_{iV}$  so that the final set contains exactly 1000 nodes.

### 3.4 Choosing $\sigma$

In Section 2.2 we introduced the  $\sigma$  parameter, and we have shown that larger values of  $\sigma$  lead to more local optima. This leaves the question of choosing the value of  $\sigma$ .

One obvious choice is  $\sigma = 0$ , which means that  $\phi_\sigma$  is exactly the classical conductance.

Another choice would be to pick the smallest  $\sigma$  that leads to a positive definite kernel. But this is a global property of the network, that is furthermore very expensive to compute.

Instead, we try several different values of  $\sigma$  for each seed, and then pick the community with the highest density, that is, the community  $C$  with the largest  $a_{CC}/|C|^2$ .

### 3.5 Exactly Recoverable Communities

We now take a brief look at which kinds of communities can be exactly recovered with gradient descent and expectation-maximization. Suppose that we wish to recover a community  $C^*$  from a seeds set  $S$ , and assume that this community is connected. Denote by  $d(i)$  the shortest path distance from a node  $i \in C^*$  to any seed node, in the subnetwork induced by  $C^*$ .

First of all, since both algorithms grow the community from the seeds, we need to look at subcommunities  $C \subseteq C^*$  centered around the seeds, by which we mean that  $d(i) \leq d(j)$  for all nodes  $i \in C$  and  $j \in C^* \setminus C$ .

Secondly, we need the community to be sufficiently densely connected to be considered a community in the first place; but at the same time the community needs to be separated from the rest of the network. Again, because the communities are grown, we require that this holds also for subcommunities that are grown from the seeds,

**Definition 3** *A community  $C^*$  is dense and isolated with threshold  $\sigma$  if for all subsets  $C \subseteq C^*$  centered around the seeds  $S$ :*

- $2a_{iC}/a_{iV} > a_{CC}/a_{CV} - \sigma$  for all nodes  $i \in C$ , and
- $2a_{iC}/a_{iV} \leq a_{CC}/a_{CV} - \sigma$  for all nodes  $i \notin C$ .

Some examples of communities that satisfy this property are cliques and quasi-cliques that are only connected to nodes of high degree.

Now denote by  $D_n$  the set of nodes  $i$  in  $C^*$  with  $d(i) \leq n$ . Clearly  $D_0 = S$ , and because the community is connected there is some  $n^*$  such that  $D_{n^*} = C^*$ .

We first look at the expectation-maximization algorithm.

**Theorem 4** *If  $C^*$  is dense and isolated, then the iterates of the EMC algorithm satisfy  $C^{(t)} = D_t$ .*

**Proof** The proof proceeds by induction. For  $t = 0$ , the only nodes  $i$  with  $d(i) = 0$  are the seeds, and  $C^{(0)} = S$  by definition.

Now suppose that  $C^{(t)} = D_t$ . Then for any node  $i$  there are three possibilities.

- $i \in D_{t+1}$ ; then because  $C^*$  is dense and  $D_t$  is centered around the seeds,  $2a_{iC^{(t)}}/a_{iV} > 1 - \phi_\sigma(C^{(t)})$ . This implies that  $\nabla\phi_\sigma(C^{(t)})_i < 0$ .
- $i \in C^* \setminus D_{t+1}$ ; then there are no edges from  $D_t$  to  $i$ , since otherwise the shortest path distance from  $i$  to a seed would be  $t + 1$ . So  $a_{iC^{(t)}} = 0$ , which implies that  $\nabla\phi_\sigma(C^{(t)})_i \geq 0$ .
- $i \notin C^*$ ; then because  $C^*$  is isolated,  $2a_{iC^{(t)}}/a_{iV} \leq 1 - \phi_\sigma(C^{(t)})$ , which implies that  $\nabla\phi_\sigma(C^{(t)})_i \geq 0$ .

hence  $\nabla\phi_\sigma(C^{(t)})_i < 0$  if  $i \in D_{t+1}$ , and  $\nabla\phi_\sigma(C^{(t)})_i \geq 0$  otherwise. This means that  $C^{(t+1)} = D_{t+1}$ . ■

For the projected gradient descent algorithm from Section 3.1 we have an analogous theorem,

**Theorem 5** *If  $C^*$  is dense and isolated, then the iterates of PGDC satisfy  $\mathbf{c}^{(t)} = [D_t]$ .*

**Proof** The proof proceeds by induction, and is analogous to the proof of Theorem 4. For  $t = 0$ , the only nodes  $i$  with  $d(i) = 0$  are the seeds, and  $\mathbf{c}^{(0)} = \mathbf{s} = [S]$  by definition.

Now suppose that  $\mathbf{c}^{(t)} = [D_t]$ . We have already shown that  $\nabla\phi_\sigma(C^{(t)})_i < 0$  if and only if  $i \in D_{t+1}$ . This means that after projecting onto the set of valid communities, only the membership of nodes in  $D_{t+1}$  can increase. Since nodes in  $D_t$  already have membership 1, and nodes not in  $D_{t+1}$  already have membership 0, they are not affected.

Let  $\gamma_{\max} = \max_{i \in D_{t+1}} -1/\nabla\phi_\sigma(\mathbf{c}^{(t)})_i$ . Clearly if  $\gamma^{(t)} \geq \gamma_{\max}$ , then  $c_i^{(t)} - \gamma^{(t)}\nabla\phi_\sigma(\mathbf{c}^{(t)})_i > 1$  for all nodes  $i \in D_{t+1}$ , and hence  $p(\mathbf{c}^{(t)} - \gamma^{(t)}\nabla\phi_\sigma(\mathbf{c}^{(t)})) = [D_{t+1}]$ . So to complete the proof, we only need to show that the optimal step size found with line search is indeed (at least)  $\gamma_{\max}$ .

Suppose that  $\gamma^{(t)} < \gamma_{\max}$  leads to the optimal conductance. Then there is a node  $i \in D_{t+1}$  with fractional membership,  $0 < c_i^{(t+1)} < 1$ . By repeated application of Theorem 2 we know that there is a discrete community  $C'$  with  $\phi_\sigma(C') = \phi_\sigma(\mathbf{c}^{(t+1)})$ , and furthermore  $\phi_\sigma(C' \setminus \{i\}) = \phi_\sigma(C' \cup \{i\})$ . The latter can only be the case if  $\nabla\phi_\sigma(C')_i = 0$ . Because the only nodes whose membership has changed compared to  $\mathbf{c}^{(t)}$  are those in  $D_{t+1} \setminus D_t$ , it follows that  $C'$  contains all nodes with distance at most  $t$  to the seeds, as well as some nodes with distance  $t + 1$  to the seeds. This means that  $C'$  is centered around the seeds, and so  $\nabla\phi_\sigma(C')_i > 0$ . This is a contradiction, which means that  $\gamma^{(t)} \geq \gamma_{\max}$  must be the optimum.  $\blacksquare$

As a corollary, since  $D_{n^*+1} = D_{n^*}$ , both the EMC and the PGDC algorithm will halt, and exactly recover  $C^*$ .

The notion of dense and isolated community is weakly related to that of  $(\alpha, \beta)$ -cluster (Mishra et al., 2008) (without the technical assumption that each node has a self-loop):  $C$  is an  $(\alpha, \beta)$ -cluster, with  $0 \leq \alpha < \beta \leq 1$  if  $a_{iC} \geq \beta|C|$  for  $i$  in  $C$ ,  $a_{iC} \leq \alpha|C|$  for  $i$  outside  $C$ .

The definition of dense and isolated community depends on the degree of the nodes while that of  $(\alpha, \beta)$ -cluster does not. As a consequence, not all maximal cliques of a graph are in general dense and isolated communities while they are  $(\alpha, \beta)$ -clusters. For instance, a maximal clique linked to an external isolated node, that is, a node of degree 1, is not dense and isolated.

In general one can easily show that if  $C$  is dense and isolated and

$$\min_{i \in C} a_{iV} > \max_{i \notin C, a_{iC} > 0} a_{iV}$$

then  $C$  is an  $(\alpha, \beta)$ -cluster with

$$\beta = \frac{1 - \phi(C)}{2|C|} \min_{i \in C} a_{iV}$$

and

$$\alpha = \frac{1 - \phi(C)}{2|C|} \max_{i \notin C, a_{iC} > 0} a_{iV}.$$

| Dataset           | #node   | #edge     | clus.c. | #comm   | $\overline{ C }$ | $\overline{\phi(C)}$ |
|-------------------|---------|-----------|---------|---------|------------------|----------------------|
| LFR (om=1)        | 5000    | 25125     | 0.039   | 101     | 49.5             | 0.302                |
| LFR (om=2)        | 5000    | 25123     | 0.021   | 146     | 51.4             | 0.534                |
| LFR (om=3)        | 5000    | 25126     | 0.016   | 191     | 52.4             | 0.647                |
| LFR (om=4)        | 5000    | 25117     | 0.015   | 234     | 53.4             | 0.717                |
| Karate            | 34      | 78        | 0.103   | 2       | 17.0             | 0.141                |
| Football          | 115     | 613       | 0.186   | 12      | 9.6              | 0.402                |
| Pol.Blogs         | 1490    | 16715     | 0.089   | 2       | 745.0            | 0.094                |
| Pol.Books         | 105     | 441       | 0.151   | 3       | 35.0             | 0.322                |
| Flickr            | 35313   | 3017530   | 0.030   | 171     | 4336.1           | 0.682                |
| Amazon            | 334863  | 925872    | 0.079   | 151037  | 19.4             | 0.554                |
| DBLP              | 317080  | 1049866   | 0.128   | 13477   | 53.4             | 0.622                |
| Youtube           | 1134890 | 2987624   | 0.002   | 8385    | 13.5             | 0.916                |
| LiveJournal       | 3997962 | 34681189  | 0.045   | 287512  | 22.3             | 0.937                |
| Orkut             | 3072441 | 117185083 | 0.014   | 6288363 | 14.2             | 0.977                |
| CYC/Gavin 2006    | 6230    | 6531      | 0.121   | 408     | 4.7              | 0.793                |
| CYC/Krogan 2006   | 6230    | 7075      | 0.075   | 408     | 4.7              | 0.733                |
| CYC/Collins 2007  | 6230    | 14401     | 0.083   | 408     | 4.7              | 0.997                |
| CYC/Costanzo 2010 | 6230    | 57772     | 0.022   | 408     | 4.7              | 0.996                |
| CYC/Hoppins 2011  | 6230    | 10093     | 0.030   | 408     | 4.7              | 0.999                |
| CYC/all           | 6230    | 80506     | 0.017   | 408     | 4.7              | 0.905                |

Table 1: Overview of the datasets used in the experiments. For each dataset we consider three different sets of communities.

## 4. Experiments

To test the proposed algorithms, we assess their performance on various networks. We also perform experiments on recent state-of-the-art algorithms based on the diffusion method which also optimize conductance.

### 4.1 Algorithms

Specifically, we perform a comparative empirical analysis of the following algorithms.

1. PGDC. The projected gradient descent algorithm for optimizing  $\sigma$ -conductance given in Algorithm 1. We show the results for two variants: PGDC-0 with  $\sigma = 0$  and PGDC-d where  $\sigma$  is chosen to maximize the community’s density as described in Section 3.4.
2. EMC. The Expectation Maximization algorithm for optimizing  $\sigma$ -conductance described in Section 2. We consider the variants EMC-0 with  $\sigma = 0$  and EMC-d where  $\sigma$  is chosen automatically.
3. YL. The algorithm by Yang and Leskovec (2012) (with conductance as scoring function), based on the diffusion method. It computes an approximation of the personalized Page Rank graph diffusion vector (Andersen et al., 2006). The values in this



vector are divided by the degree of the corresponding nodes, and the nodes are sorted in descending order by their values. The ranking induces a one dimensional search space of communities  $C_k$ , called a sweep, defined by the sequence of prefixes of the sorted list, that is, the  $k$  top ranked nodes, for  $k = 1, \dots, |V|$ . The smallest  $k$  whose  $C_k$  is a ‘local optimum’ of conductance is computed and  $C_k$  is extracted. Local optima of conductance over the one dimensional space  $C_1, C_2, \dots, C_{|V|}$  are computed using a heuristic. For increasing  $k = 1, 2, \dots$   $\phi(C_k)$  is measured. When  $\phi(C_k)$  stops decreasing at  $k^*$  this is a ‘candidate point’ for a local minimum. It becomes a selected local minimum if  $\phi(C_k)$  keeps increasing after  $k^*$  and eventually becomes higher than  $\alpha\phi(C_k)$ , otherwise it is discarded.  $\alpha = 1.2$  is shown to give good results and is also used in our experiments. Yang and Leskovec (2012) show that finding the local optima of the sweep curve instead of the global optimum gives a large improvement over previous local spectral clustering methods by Andersen and Lang (2006) and by Spielman and Teng (2004).

4. HK. The algorithm by Kloster and Gleich (2014), also based on the diffusion method. Here, instead of using the Personalized PageRank score, nodes are ranked based on a Heat Kernel diffusion score (Chung, 2007). We use the implementation made available by Kloster and Gleich (2014), which tries different values of the algorithm’s parameters  $t$  and  $\epsilon$ , and picks the community with the highest conductance among them. The details are in section 6.2 of (Kloster and Gleich, 2014). Code is available at <https://www.cs.purdue.edu/homes/dgleich/codes/hkgrow>.
5. PPR. The pprpush algorithm by Andersen and Lang (2006) based on the personalized Page Rank graph diffusion. Compared to YL instead of finding a local optimum of the sweep, the method looks for a global optimum, and hence often finds larger communities. We use the implementation included with the HK method.

## 4.2 Datasets

### 4.2.1 ARTIFICIAL DATASETS

The first set of experiments we performed is on artificially generated networks with a known community structure. We use the LFR benchmark (Lancichinetti et al., 2008). We used the parameter settings  $N=5000$   $\mu=0.3$   $k=10$   $\maxk=50$   $t1=2$   $t2=1$   $\minc=20$   $\maxc=100$   $on=2500$ , which means that the graph has 5000 nodes, and between 20 and 100 communities, each with between 10 and 50 nodes. Half of the nodes, 2500 are a member of multiple communities. We vary the overlap parameter ( $om$ ), which determines how many communities these nodes are in. More overlap makes the problem harder.

### 4.2.2 SOCIAL AND INFORMATION NETWORK DATASETS WITH GROUND TRUTH

We use five social and information network datasets with ground-truth from the SNAP collection (Leskovec and Krevl, 2014). These datasets are summarized in Table 1. For each dataset we list the number of nodes, number of edges and the clustering coefficient. We consider all available ground truth communities with at least 3 nodes.

Yang and Leskovec (2012) also defined a set of top 5000 communities for each dataset. These are communities with a high combined score for several community goodness metrics,

among which is conductance. We therefore believe that communities in this set are biased to be more easy to recover by optimizing conductance, and therefore do not consider them here. Results with these top 5000 ground truth communities are available in tables 1–3 in the supplementary material <sup>2</sup>.

In addition to the SNAP datasets we also include the Flickr social network dataset (Wang et al., 2012).

#### 4.2.3 PROTEIN INTERACTION NETWORK DATASETS

We have also run experiments on protein interaction networks of yeast from the BioGRID database (Stark et al., 2006). This database curates networks from several different studies. We have constructed networks for Gavin et al. (2006), Krogan et al. (2006), Collins et al. (2007), Costanzo et al. (2010), Hoppins et al. (2011), as well as a network that is the union of all interaction networks confirmed by physical experiments.

As ground truth communities we take the CYC2008 catalog of protein complexes for each of the networks (Pu et al., 2009).

#### 4.2.4 OTHER DATASETS

Additionally we used some classical datasets with known communities: Zachary’s karate club Zachary (1977); Football: A network of American college football games (Girvan and Newman, 2002); Political books: A network of books about US politics (Krebs, 2004); and Political blogs: Hyperlinks between weblogs on US politics (Adamic and Glance, 2005). These datasets might not be very well suited for this problem, since they have very few communities.

### 4.3 Results

In all our experiments we use a single seed node, drawn uniformly at random from the community. We have also performed experiments with multiple seeds; the results of those experiments can be found in the supplementary material.

To keep the computation time manageable we have performed all experiments on a random sample of 1000 ground-truth communities. For datasets with fewer than 1000 communities, we include the same community multiple times with different seeds.

Since the datasets here considered have information about ground truth communities, a natural external validation criterion to assess the performance of algorithms on these datasets is to compare the community produced by an algorithm with the ground truth one. In general, that is, when ground truth information is not available, this task is more subtle, because it is not clear what is a good external validation metric to evaluate a community (Yang and Leskovec, 2015).

We measure quality performance with the  $F_1$  score, which for community finding can be defined as

$$F_1(C, C^*) = 2 \frac{|C \cap C^*|}{|C| + |C^*|},$$

---

2. The supplementary material is available from <http://cs.ru.nl/~tvanlaarhoven/conductance2016>

| Dataset           | PGDC-0       | PGDC-d       | EMc-0        | EMc-d        | YL           | HK           | PPR          |
|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| LFR (om=1)        | <b>0.967</b> | 0.185        | 0.868        | 0.187        | 0.203        | 0.040        | 0.041        |
| LFR (om=2)        | <b>0.483</b> | 0.095        | 0.293        | 0.092        | 0.122        | 0.039        | 0.041        |
| LFR (om=3)        | <b>0.275</b> | 0.085        | 0.158        | 0.083        | 0.110        | 0.037        | 0.039        |
| LFR (om=4)        | <b>0.178</b> | 0.074        | 0.100        | 0.072        | 0.092        | 0.032        | 0.034        |
| Karate            | 0.831        | 0.472        | 0.816        | 0.467        | 0.600        | 0.811        | <b>0.914</b> |
| Football          | <b>0.792</b> | <b>0.816</b> | 0.766        | <b>0.805</b> | <b>0.816</b> | 0.471        | 0.283        |
| Pol.Blogs         | <b>0.646</b> | 0.141        | <b>0.661</b> | 0.149        | 0.017        | <b>0.661</b> | 0.535        |
| Pol.Books         | 0.596        | 0.187        | 0.622        | 0.197        | 0.225        | <b>0.641</b> | <b>0.663</b> |
| Flickr            | 0.098        | 0.027        | 0.097        | 0.027        | 0.013        | 0.054        | <b>0.118</b> |
| Amazon            | 0.470        | <b>0.522</b> | 0.425        | <b>0.522</b> | 0.493        | 0.245        | 0.130        |
| DBLP              | <b>0.356</b> | <b>0.369</b> | 0.317        | <b>0.371</b> | 0.341        | 0.214        | 0.210        |
| Youtube           | 0.089        | <b>0.251</b> | 0.073        | <b>0.248</b> | 0.228        | 0.037        | 0.071        |
| LiveJournal       | 0.067        | <b>0.262</b> | 0.059        | <b>0.259</b> | 0.183        | 0.035        | 0.049        |
| Orkut             | 0.042        | <b>0.231</b> | 0.033        | <b>0.231</b> | 0.171        | 0.057        | 0.033        |
| CYC/Gavin 2006    | 0.474        | <b>0.543</b> | 0.455        | <b>0.543</b> | <b>0.526</b> | 0.336        | 0.294        |
| CYC/Krogan 2006   | 0.410        | <b>0.513</b> | 0.364        | <b>0.511</b> | <b>0.504</b> | 0.229        | 0.169        |
| CYC/Collins 2007  | 0.346        | <b>0.429</b> | 0.345        | <b>0.429</b> | <b>0.416</b> | 0.345        | 0.345        |
| CYC/Costanzo 2010 | 0.174        | <b>0.355</b> | 0.172        | <b>0.351</b> | 0.314        | 0.170        | 0.170        |
| CYC/Hoppins 2011  | 0.368        | <b>0.405</b> | 0.368        | <b>0.405</b> | <b>0.424</b> | 0.368        | 0.368        |
| CYC/all           | 0.044        | <b>0.459</b> | 0.017        | <b>0.459</b> | 0.425        | 0.016        | 0.002        |

Table 2: Average  $F_1$  score between recovered communities and ground-truth. The best result for each dataset is indicated in bold, as are the results not significantly worse according to a paired T-test (at significance level 0.01).

where  $C$  is the recovered community and  $C^*$  is the ground truth one. A higher  $F_1$  score is better, with 1 indicating a perfect correspondence between the two communities.

Note that a seed node might be in multiple ground truth communities. In this case we only compare the recovered community to the true community that we started with. If a method finds another ground truth community this is not detected, and so it results in a low  $F_1$  score.

We also analyze the output of these algorithms with respect to the conductance of produced communities and their size. Results on the run time of the algorithms are reported in the supplementary material (Table 4).

Figure 2 shows the  $F_1$  scores as a function of the parameter  $\sigma$ . Table 2 shows the  $F_1$  scores comparing the results of the methods to the true communities. Table 3 shows the mean size of the found communities, and Table 4 their conductance.

In general, results of these experiments indicate that on real-life networks, our methods based on continuous relaxation of conductance, PPR and HK produce communities with good conductance, but all are less faithful to the ground truth when the network contains many small communities. In PGDC, EMC the automatic choice of  $\sigma$  helps to achieve results closer to the ground truth, and the built-in tendency of YL to favor small communities helps

| Dataset           | PGDC-0 | PGDC-d | EMc-0 | EMc-d | YL   | HK     | PPR     |
|-------------------|--------|--------|-------|-------|------|--------|---------|
| LFR (om=1)        | 52.3   | 6.2    | 71.8  | 6.3   | 5.9  | 2410.0 | 2366.2  |
| LFR (om=2)        | 93.2   | 5.6    | 292.8 | 6.4   | 4.9  | 2404.4 | 2311.6  |
| LFR (om=3)        | 104.7  | 5.6    | 451.5 | 6.4   | 5.0  | 2399.4 | 2283.7  |
| LFR (om=4)        | 108.9  | 4.9    | 530.2 | 5.7   | 4.9  | 2389.1 | 2262.0  |
| Karate            | 20.0   | 8.0    | 24.1  | 8.0   | 8.8  | 16.7   | 17.1    |
| Football          | 14.7   | 9.5    | 16.2  | 9.4   | 8.8  | 40.5   | 56.5    |
| Pol.Blogs         | 515.6  | 110.1  | 538.9 | 118.1 | 7.2  | 492.7  | 1051.1  |
| Pol.Books         | 37.8   | 5.2    | 43.1  | 5.7   | 6.7  | 49.3   | 53.4    |
| Flickr            | 639.9  | 73.0   | 644.6 | 73.5  | 12.9 | 174.2  | 1158.1  |
| Amazon            | 25.2   | 5.6    | 45.6  | 5.7   | 6.4  | 88.8   | 20819.9 |
| DBLP              | 61.9   | 5.4    | 83.5  | 6.1   | 6.0  | 55.0   | 24495.0 |
| Youtube           | 340.6  | 19.4   | 474.2 | 21.3  | 9.3  | 147.9  | 20955.5 |
| LiveJournal       | 243.7  | 5.5    | 309.3 | 5.9   | 10.8 | 153.2  | 3428.7  |
| Orkut             | 245.1  | 17.9   | 344.7 | 19.1  | 11.1 | 212.0  | 1634.0  |
| CYC/Gavin 2006    | 19.7   | 3.4    | 34.3  | 3.5   | 3.1  | 236.7  | 621.9   |
| CYC/Krogan 2006   | 48.4   | 7.0    | 138.8 | 9.4   | 3.7  | 723.8  | 1756.3  |
| CYC/Collins 2007  | 202.9  | 19.5   | 207.2 | 19.5  | 2.6  | 192.0  | 189.4   |
| CYC/Costanzo 2010 | 540.2  | 58.1   | 564.2 | 66.5  | 5.9  | 1058.8 | 942.9   |
| CYC/Hoppins 2011  | 229.9  | 110.1  | 235.5 | 110.4 | 4.3  | 295.2  | 295.2   |
| CYC/all           | 657.5  | 16.0   | 841.9 | 17.0  | 9.6  | 2795.2 | 5786.0  |

Table 3: Average size of the recovered communities.

as well. On the other hand, on networks with large communities our methods, PPR and HK work best. On the artificial LFR data continuous relaxation of conductance seems to work best. This result indicates that the LFR model of ‘what is a community’ is somehow in agreement with the notion of local community as local optimum of the continuous relaxation of conductance. However, as observed in recent works like (Jeub et al., 2015), the LFR model does not seem to represent the diverse characteristics of real-life communities.

We have included tables of the standard deviation in the supplementary material. Overall, the standard deviation in cluster size is of the same order of magnitude as the mean. The standard deviation of the conductance is around 0.1 for LFR datasets, 0.2 for the SNAP datasets and 0.3 for the CYC datasets. It is not surprising that the variance is this high, because the communities vary a lot in size and density.

Results on these datasets can be summarized as follows.

#### 4.3.1 ARTIFICIAL LFR DATASETS

On these datasets, HK and PPR tend to find communities that are much too large, with small conductance but also with low  $F_1$  scores. This happens because the LFR networks are small, and the methods are therefore able to consider a large part of the nodes in the network.

| Dataset           | PGDC-0       | PGDC-d | EMc-0        | EMc-d | YL    | HK           | PPR          |
|-------------------|--------------|--------|--------------|-------|-------|--------------|--------------|
| LFR (om=1)        | 0.301        | 0.750  | 0.304        | 0.749 | 0.755 | <b>0.250</b> | 0.273        |
| LFR (om=2)        | 0.532        | 0.786  | 0.541        | 0.787 | 0.780 | <b>0.315</b> | 0.338        |
| LFR (om=3)        | 0.589        | 0.793  | 0.587        | 0.793 | 0.781 | <b>0.333</b> | 0.354        |
| LFR (om=4)        | 0.604        | 0.791  | 0.595        | 0.792 | 0.775 | <b>0.341</b> | 0.359        |
| Karate            | 0.129        | 0.460  | <b>0.081</b> | 0.475 | 0.327 | 0.222        | 0.136        |
| Football          | 0.277        | 0.356  | 0.274        | 0.362 | 0.385 | 0.244        | <b>0.155</b> |
| Pol.Blogs         | 0.228        | 0.743  | 0.212        | 0.737 | 0.867 | 0.229        | <b>0.137</b> |
| Pol.Books         | 0.140        | 0.622  | 0.107        | 0.611 | 0.571 | 0.127        | <b>0.065</b> |
| Flickr            | <b>0.777</b> | 0.937  | <b>0.777</b> | 0.937 | 0.951 | 0.864        | <b>0.762</b> |
| Amazon            | 0.181        | 0.464  | 0.180        | 0.463 | 0.402 | 0.081        | <b>0.053</b> |
| DBLP              | 0.246        | 0.571  | 0.257        | 0.565 | 0.498 | <b>0.133</b> | 0.147        |
| Youtube           | 0.601        | 0.765  | 0.711        | 0.759 | 0.700 | <b>0.201</b> | 0.341        |
| LiveJournal       | 0.563        | 0.875  | 0.589        | 0.874 | 0.774 | <b>0.336</b> | 0.489        |
| Orkut             | <b>0.718</b> | 0.916  | 0.731        | 0.917 | 0.928 | 0.750        | <b>0.711</b> |
| CYC/Gavin 2006    | 0.614        | 0.734  | 0.611        | 0.732 | 0.735 | <b>0.532</b> | <b>0.500</b> |
| CYC/Krogan 2006   | 0.466        | 0.626  | 0.469        | 0.620 | 0.617 | 0.325        | <b>0.265</b> |
| CYC/Collins 2007  | <b>0.716</b> | 0.953  | <b>0.712</b> | 0.953 | 0.972 | <b>0.720</b> | <b>0.730</b> |
| CYC/Costanzo 2010 | 0.759        | 0.931  | 0.755        | 0.929 | 0.934 | <b>0.672</b> | <b>0.646</b> |
| CYC/Hoppins 2011  | <b>0.788</b> | 0.883  | <b>0.785</b> | 0.882 | 0.970 | <b>0.763</b> | <b>0.763</b> |
| CYC/all           | 0.674        | 0.872  | 0.742        | 0.874 | 0.840 | 0.363        | <b>0.026</b> |

Table 4: Average conductance of the recovered communities.

On the other hand, YL always starts its search at small communities, and it stops early, so the communities it finds are smaller than the ground truth ones on these networks.

The best  $F_1$  results are achieved by PGDC with  $\sigma = 0$ , that is, when the continuous relaxation of conductance is used as the objective function. This method employs a more powerful optimizer than YL, so it is able to find a large community with a better conductance, but it still stops at the first local optimum. In the LFR datasets these optima are very clear, and correspond closely to the ground truth communities.

In all cases EMC shows similar or slightly worse performance compared to PGDC, so the gradient descend algorithm should be preferred.

The automatic choice of  $\sigma$  leads to communities which are of relatively small size. We believe that this happens because the nodes in LFR datasets all have exactly the same fraction of within community edges. Increasing  $\sigma$  suddenly makes the gradient for most of these nodes positive. In real networks there are often hubs that are more central to a community, with more connections to the community’s nodes and to the seed. These hubs still can be found at higher values of  $\sigma$ .

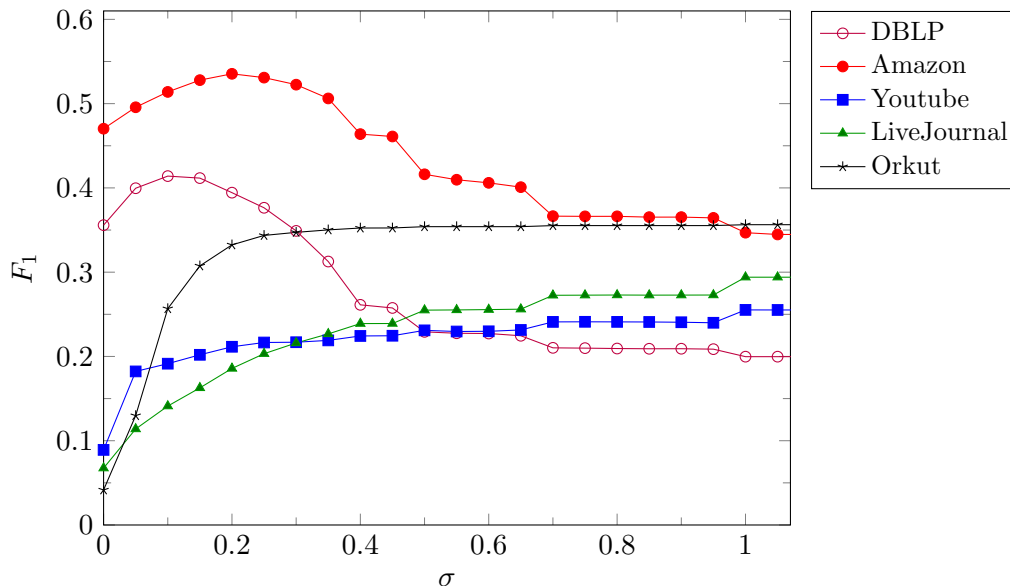


Figure 2: Average  $F_1$  score as a function of the  $\sigma$  parameter on the SNAP datasets with the PGDC method.

#### 4.3.2 SMALL REAL-LIFE SOCIAL NETWORKS WITH FEW COMMUNITIES (KARATE, FOOTBALL, BLOGS, BOOKS)

Our methods based on continuous relaxation of conductance yield the best  $F_1$  results on the Football and Blog networks, while PPR performs best on the other two networks and achieves best overall conductance.

#### 4.3.3 LARGE SOCIAL NETWORK WITH BIG COMMUNITIES (FLICKR)

On this network, PPR achieves the best results both in terms of  $F_1$  score as well as conductance. However the the produced communities are about four time smaller than the ground truth communities, which have more than 4000 nodes. PGDC and EMC with  $\sigma = 0$  yield communities of conductance similar to that of PPR communities, but their size is smaller (about 650 nodes). This happens because the algorithms are restricted to 1000 nodes around the seed, without this restriction larger communities would be found. Somewhat surprisingly HK produces communities of relatively small size (about 175 nodes). The automatic choice of  $\sigma$  yields to even smaller communities (about 64 nodes). The smallest size communities are produced by YL (about 13 nodes).

#### 4.3.4 LARGE REAL-LIFE SNAP NETWORKS WITH MANY SMALL COMMUNITIES

On these networks the automatic choice of  $\sigma$  gives best results, consistently outperforming the other algorithms. In Figure 2, the  $F_1$  score of PGDC and EMC as a function of  $\sigma$  is plotted. For some datasets a small value of  $\sigma$  works well, while for others a larger value of  $\sigma$  is better. Our procedure to choose  $\sigma$  produces results that are close to, but slightly below,

the best a posteriori choice of  $\sigma$ . So on these networks the proposed procedure positively affects the performance of our algorithms. YL favors communities of small size less faithful to the ground truth. PGDC-0, EMC-0, PPR and HK ‘explode’, and produce very large communities. For our methods this ‘explosion’ is limited only because we limit the search to 1000 nodes near the seed. The ground-truth communities of these datasets have rather high conductance, and the networks have a very low clustering coefficient. In such a case, communities have many links to nodes outside, hence conductance alone is clearly not suited to finding these type of local communities.

#### 4.3.5 REAL-LIFE PROTEIN INTERACTION NETWORKS WITH VERY SMALL COMMUNITIES (CYC)

Also on these networks the automatic choice of  $\sigma$  gives best results. As expected, due to the very small size of the ground truth communities, YL also achieves very good results. The other algorithms tend to produce less realistic, large communities which have better conductance.

#### 4.3.6 RUNNING TIME

The best performing algorithm with respect to running time is HK. PGDC and EMC are about four times slower with a fixed value of  $\sigma$ , and ten to twenty times slower when automatically determining  $\sigma$ . The running times are included in Table 4 of the supplementary material. All experiments were run on a 2.4GHz Intel XEON E7-4870 machine. Note that the different methods are implemented in different languages (our implementation is written in Octave, while HK and PPR are implemented in C++), so the running times only give an indication of the overall trend, and can not be compared directly.

#### 4.3.7 TOP 5000 COMMUNITIES

Results with only the top 5000 ground truth communities available at the SNAP dataset collection are similar to the results with all communities. As expected, the  $F_1$  score is much higher and the conductance of the recovered community is better. Because these ground truth communities have a better conductance, it is better to optimize conductance, that is to take  $\sigma = 0$ . As a consequence the performance of PGDC-0 and EMC-0 is better than that of PGDC-d and EMC-d for these communities. The full results are available in Tables 1–3 in the supplementary material.

## 5. Discussion

This paper investigated conductance as an objective function for local community detection from a set of seeds. By making a continuous relaxation of conductance we show how standard techniques such as projected gradient descent can be used to optimize it. Even though this is a continuous optimization problem, we show that the local optima are almost always discrete communities. We further showed how linking conductance with kernel weighted  $k$ -means clustering leads to the new  $\sigma$ -conductance objective function and to simple yet effective algorithms for local community detection by seed expansion.

We provided a formalization of a class of good local communities around a set of seeds and showed that the proposed algorithms can find them. We suspect that these communities can also be exactly retrieved using local community algorithms based on the diffusion method, but do not yet have a proof. The condition that such communities should be centered around the seeds raises the question of how to find such seeds. Although various works have studied seed selection for diffusion based algorithms, such as Kloumann and Kleinberg (2014), this problem remains to be investigated in the context of local community detection by  $\sigma$ -conductance optimization using PGDC and EMC.

Our experimental results indicate the effectiveness of direct optimization of a continuous relaxation of  $\sigma$ -conductance using gradient descent and expectation maximization. In our algorithms we used community density as a criterion to choose  $\sigma$ . This resulted to be a good choice for the performance of our algorithms on the SNAP networks. It would be interesting to investigate also other criteria to choose  $\sigma$ . Conversely, the fact that maximum density is a good criterion for selecting  $\sigma$  implies that it might also be directly optimized as an objective for finding communities.

On some datasets, when optimizing normal conductance, that is, with  $\sigma = 0$ , our methods sometimes find very large communities. These communities will have a very good conductance, but they do not correspond well to the ground truth. In some sense the optimizer is ‘too good’, and conductance is not the best criterion to describe these communities. A better objective would perhaps take into account the size of the community more explicitly, but this needs to be investigated further.

In this paper we have used gradient descent, a first order optimization method which utilizes only the objective function’s gradient. More advanced optimization methods also use second derivatives or approximations of those. We believe that such methods will not bring a large advantage compared to gradient descent, because during the optimization many coordinates are at the boundary value 0 or 1, and second derivatives would not help to locate these boundary points. Other constrained optimizers such as interior point methods have the problem that they need to inspect a much larger part of the network, potentially all of it, because intermediate steps have nonzero membership for all nodes.

## Acknowledgments

We thank the reviewers for their useful comments. This work has been partially funded by the Netherlands Organization for Scientific Research (NWO) within the EW TOP Compartment 1 project 612.001.352.

## References

- Lada Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.
- Reid Andersen and Kevin J. Lang. Communities from seed sets. In *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, pages 223–232, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9.



- Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.
- Haim Avron and Lior Horesh. Community detection using time-dependent personalized pagerank. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1795–1803, 2015.
- Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity, CCC '05*, pages 144–153, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2364-1.
- Jiyang Chen, Osmar Zaïane, and Randy Goebel. Local community identification in social networks. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pages 237–242. IEEE, 2009.
- Qiong Chen, Ting-Ting Wu, and Ming Fang. Detecting local community structures in complex networks based on local degree central nodes. *Physica A: Statistical Mechanics and its Applications*, 392(3):529–537, 2013.
- Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.
- Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2):026132, 2005.
- Sean Collins, et al. Towards a comprehensive atlas of the physical interactome of *saccharomyces cerevisiae*. *Molecular Cellular Proteomics*, pages 600381–600200, 2007.
- Michael Costanzo, et al. The genetic landscape of a cell. *Science*, 327(5964):425–431, 2010.
- Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, November 2007. ISSN 0162-8828.
- Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
- Ullas Gargi, Wenjun Lu, Vahab S Mirrokni, and Sangho Yoon. Large-scale community detection on youtube for topic discovery and exploration. In *ICWSM*, 2011.
- Anne-Claude Gavin, et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.
- Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

- David F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 597–605, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6.
- Suzanne Hoppins, et al. A mitochondrial-focused genetic interaction map reveals a scaffold-like complex required for inner membrane organization in mitochondria. *The Journal of Cell Biology*, 195(2):323–340, 2011.
- Lucas GS Jeub, Prakash Balachandran, Mason A Porter, Peter J Mucha, and Michael W Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *PHYSICAL REVIEW E Phys Rev E*, 91:012821, 2015.
- U Kang and Christos Faloutsos. Beyond “caveman communities”: Hubs and spokes for graph compression and mining. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 300–309. IEEE, 2011.
- Kyle Kloster and David F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1386–1395, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9.
- Isabel M Kloumann and Jon M Kleinberg. Community membership identification from small seed sets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1366–1375. ACM, 2014.
- Valdis Krebs. New political patterns. Editorial, 2004.
- Nevan J. Krogan, et al. Global landscape of protein complexes in the yeast *saccharomyces cerevisiae*. *Nature*, 440(7084):637–643, 2006.
- Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 2008.
- Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, Santo Fortunato, et al. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.
- Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.
- Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.
- László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 346–354. IEEE, 1990.

- Michael W Mahoney, Lorenzo Orecchia, and Nisheeth K Vishnoi. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *The Journal of Machine Learning Research*, 13(1):2339–2365, 2012.
- Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert Endre Tarjan. Finding strongly knit clusters in social networks. *Internet Mathematics*, 5(1):155–174, 2008.
- Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 251–260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6.
- Shuye Pu, Jessica Wong, Brian Turner, Emerson Cho, and Shoshana J. Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 37(3):825–831, 2009.
- Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007. ISSN 15740137.
- Jiří Šíma and Satu Elisa Schaeffer. On the NP-completeness of some graph cluster measures. In *SOFSEM 2006: Theory and Practice of Computer Science*, pages 530–537. Springer, 2006.
- Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2004.
- Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(Database-Issue):535–539, 2006.
- Xufei Wang, Lei Tang, Huan Liu, and Lei Wang. Learning with multi-resolution overlapping communities. *Knowledge and Information Systems (KAIS)*, 2012.
- Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2099–2108. ACM, 2013.
- Xiao-Ming Wu, Zhenguo Li, Anthony M So, John Wright, and Shih-Fu Chang. Learning with partially absorbing random walks. In *Advances in Neural Information Processing Systems*, pages 3077–3085, 2012.
- Yubao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. Robust local community detection: On free rider effect and its elimination. *Proc. VLDB Endow.*, 8(7):798–809, February 2015. ISSN 2150-8097.
- Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, MDS '12*, pages 3:1–3:8. ACM, 2012. ISBN 978-1-4503-1546-3.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. ISSN 0219-1377.

Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

Zeyuan A Zhu, Silvio Lattanzi, and Vahab Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 396–404, 2013a.

Zeyuan Allen Zhu, Silvio Lattanzi, and Vahab Mirrokni. Local graph clustering beyond Cheeger’s inequality. *arXiv preprint arXiv:1304.8132*, 2013b.