Empirical Comparison of Algorithms for Network Community Detection

Jure Leskovec  
Stanford University  
jure@cs.stanford.edu

Kevin J. Lang  
Yahoo! Research  
langk@yahoo-inc.com

Michael W. Mahoney  
Stanford University  
mmahoney@cs.stanford.edu

ABSTRACT

Detecting clusters or communities in large real-world graphs such as large social or information networks is a problem of considerable interest. In practice, one typically chooses an objective function that captures the intuition of a network cluster as a set of nodes with better internal connectivity than external connectivity, and then one applies approximation algorithms or heuristics to extract sets of nodes that are related to the objective function and that “look like” good communities for the application of interest.

In this paper, we explore a range of network community detection methods in order to compare them and to understand their relative performance and the systematic biases in the clusters they identify. We evaluate several common objective functions that are used to formalize the notion of a network community, and we examine several different classes of approximation algorithms that aim to optimize such objective functions. In addition, rather than simply fixing an objective and asking for an approximation to the best cluster of any size, we consider a size-resolved version of the optimization problem. Considering community quality as a function of its size provides a much finer lens with which to examine community detection algorithms, since objective functions and approximation algorithms often have non-obvious size-dependent behavior.

Categories and Subject Descriptors: H.2.8 Database Management: Database applications – Data mining

General Terms: Measurement; Experimentation.

Keywords: Community structure; Graph partitioning; Conductance; Spectral methods; Flow-based methods.

1. INTRODUCTION

Detecting clusters or communities in real-world graphs such as large social networks, web graphs, and biological networks is a problem of considerable practical interest that has received a great deal of attention [16, 17, 13, 8, 19]. A “network community” (also sometimes referred to as a module or cluster) is typically thought of as a group of nodes with more and/or better interactions amongst its members than between its members and the remainder of the network [30, 16].

To extract such sets of nodes one typically chooses an objective function that captures the above intuition of a community as a set of nodes with better internal connectivity than external connectivity. Then, since the objective is typically NP-hard to optimize exactly [24, 4, 31], one employs heuristics [16, 20, 9] or approximation algorithms [25, 33, 2] to find sets of nodes that approximately optimize the objective function and that can be understood or interpreted as “real” communities. Alternatively, one might define communities operationally to be the output of a community detection procedure, hoping they bear some relationship to the intuition as to what it means for a set of nodes to be a good community [16, 29]. Once extracted, such clusters of nodes are often interpreted as organizational units in social networks, functional units in biochemical networks, ecological niches in food web networks, or scientific disciplines in citation and collaboration networks [16, 30].

In applications, it is important to note that heuristic approaches to and approximation algorithms for community detection often find clusters that are systematically “biased,” in the sense that they return sets of nodes with properties that might be substantially different than the set of nodes that achieves the global optimum of the chosen objective. For example, many spectral-based methods tend to find compact clusters at the expense that they are not so well separated from the rest of the network; while other methods tend to find better-separated clusters that may internally be “less nice.” Moreover, certain methods tend to perform particularly well or particularly poorly on certain kinds of graphs, e.g., low-dimensional manifolds or expanders. Thus, drawing on this experience, it is of interest to compare these algorithms on large real-world networks that have many complex structural features such as sparsity, heavy-tailed degree distributions, small diameters, etc. Moreover, depending on the particular application and the properties of the network being analyzed, one might prefer to identify specific types of clusters. Understanding structural properties of clusters identified by various algorithmic methods and various objective functions can guide in selecting the most appropriate graph clustering method in the context of a given network and target application.

In this paper, we explore a range of different community detection methods in order to elucidate these issues and to understand better the performance and biases of various network community detection algorithms on different kinds of networks. To do so, we consider a set of more than 40 networks; 12 common objective functions that are used to formalize the concept of community quality; and 8 different classes of approximation algorithms to find network communities. One should note that we are not primarily interested in finding the “best” community detection method or the most “realistic” formalization of a network community. Instead, we aim to understand the structural properties of clusters identified by various methods, and then depending on the particular application one could choose the most suitable clustering method.

We describe several classes of empirical evaluations of methods for network community detection to demonstrate the artificial properties and systematic biases of various community detection objective functions and approximation algorithms. We also discuss several meta-issues related to community detection algo-
rithms in very large graphs, including whether or not existing algorithms are sufficiently powerful to recover interesting communities and whether or not meaningful communities exist at all. Also in contrast to previous attempts to evaluate community detection algorithms and/or objective functions, we consider a size-resolved version of the typical optimization problem. That is, rather than simply fixing an objective and asking for an approximation to the best cluster of any size or some fixed partitioning, we ask for an approximation to the best cluster for every possible size. This provides a much finer lens with which to examine community detection algorithms, since objective functions and approximation algorithms often have non-obvious size-dependent behavior.

The rest of the paper is organized as follows. Section 2 gives the background and surveys the rich related work in the area of network community detection. Then, in Section 3.1, we compare structural properties of clusters extracted by two clustering methods based on two completely different computational paradigms—a spectral-based graph partitioning method Local Spectral and a flow-based partitioning algorithm Metis+MQI; and in Section 3.2, we extend the analyses by considering related heuristic-based clustering algorithms that in practice perform very well. Section 4 then focuses on 11 different objective functions that attempt to capture the notion of a community as a set of nodes with better intra- than inter-connectivity. To understand the performance of various community detection algorithms at different size scales we compute theoretical lower bounds on the conductance community-quality score in Section 5. We conclude in Section 6 with some general observations.

2. RELATED WORK AND BACKGROUND

Here we survey related work and summarize our previous work, with an emphasis on technical issues that motivate this paper.

2.1 Related work

A great deal of work has been devoted to finding communities in large networks, and much of this has been devoted to formalizing the intuition that a community is a set of nodes that has more and/or better links between its members than with the remainder of the network. Very relevant to our work is that of Kannan, Vempala, and Vetta [18], who analyze spectral algorithms and describe a community concept in terms of a bicriterion depending on the conductance of the communities and the relative weight of between-community edges. Flake, Tarjan, and Tsioutsouliklis [11] introduce a similar bicriterion that is based on network flow ideas, and Flake et al. [10] defined a community as a set of nodes that has more edges pointing inside the community than to the rest of the network. Similar edge-counting ideas were used by Radicchi et al. [30] to define and apply the notions of a strong community and a weak community.

Within the “complex networks” community, Girvan and Newman [16] proposed an algorithm that used “betweenness centrality” to find community boundaries. Following this, Newman and Girvan [19] introduced modularity to find community boundaries. Following this, Newman and Girvan [16] proposed an algorithm that used “betweenness centrality” ideas were used by Radicchi [30] to define and a posteriori measure of the quality of a single cluster as a function of its size. Moreover, the NCP also provides a finer lens with which to examine community detection within the “complex networks” community, Girvan and Newman [16] showed that characterizes the quality of network communities as a function of their size. For every $k$ between 1 and half the number of nodes in the network, we define $\Phi(k) = \min_{|S|=k} f(S)$. That is, for every possible community size $k$, $f(k)$ measures the score of the most community-like set of nodes of that size, and the NCP measures $\Phi(k)$ as a function of $k$.

For example, in Figure 1(middle) we use conductance as a measure of cluster quality and for $k = 4$, among all sets of 4-nodes, $B$ has best conductance, and thus $\Phi(4) = \frac{1}{17}$. Similarly, $D$ and $D+E$ denote the best conductance sets on 3 and 6 nodes, respectively. Just as the magnitude of the conductance provides information about how community-like is a set of nodes, the shape of the NCP provides insight into how well expressed are network communities as a function of their size. Moreover, the NCP also provides a lens to examine the quality of clusters of various sizes. Thus in the majority of our experiments we will examine and compare differ-

1Networks used in this paper are available at the supporting website [1]: http://snap.stanford.edu/narp.

2Note that one only needs to consider clusters of sizes up to half the number of nodes in the network since $\phi(S) = \phi(V \setminus S)$.

See [1] for further information and properties of these networks. Very even though we consider various notions of community score we will primarily work with conductance, which arguably is the simplest notion of cluster quality, as it can be simply thought of as the ratio between the number of edges inside the cluster and the number of edge leaving the cluster [32, 18]. More formally, conductance $\phi(S)$ of a set of nodes $S$ is $\phi(S) = c_S / \min\{Vol(S), Vol(V \setminus S)\}$, where $c_S$ denotes the size of the edge boundary, $c_S = \{|(u,v) : u \in S, v \notin S\}$, and $Vol(S) = \sum_{u \in S} d(u)$, where $d(u)$ is the degree of node $u$. Thus, in particular, more community-like sets of nodes have lower conductance. For example in Figure 1(left), sets $A$ and $B$ have conductance $\phi(A) = \frac{14}{11} > \phi(B) = \frac{7}{5}$, so the set of nodes $B$ is more community-like than the set $A$. Conductance captures a notion of “surface area-to-volume,” and thus it is widely-used to capture quantitatively the gestalt notion of a good network community as a set of nodes that has better internal- than external-connectivity [14, 31].

We then generalize the notion of the quality of a single cluster into a size resolved version. Using a particular measure of network community quality $f(S)$, e.g., conductance or one of the other measures described in Section 4, we then define the network community profile (NCP) [27, 26] that characterizes the quality of network communities as a function of their size. For every $k$ between 1 and half the number of nodes in the network, we define $\Phi(k) = \min_{|S|=k} f(S)$. That is, for every possible community size $k$, $f(k)$ measures the score of the most community-like set of nodes of that size, and the NCP measures $\Phi(k)$ as a function of $k$.

For example, in Figure 1(middle) we use conductance as a measure of cluster quality and for $k = 4$, among all sets of 4-nodes, $B$ has best conductance, and thus $\Phi(4) = \frac{1}{17}$. Similarly, $D$ and $D+E$ denote the best conductance sets on 3 and 6 nodes, respectively. Just as the magnitude of the conductance provides information about how community-like is a set of nodes, the shape of the NCP provides insight into how well expressed are network communities as a function of their size. Moreover, the NCP also provides a lens to examine the quality of clusters of various sizes. Thus in the majority of our experiments we will examine and compare differ-

### Footnote:

1Networks used in this paper are available at the supporting website [1]: http://snap.stanford.edu/narp.

2Note that one only needs to consider clusters of sizes up to half the number of nodes in the network since $\phi(S) = \phi(V \setminus S)$. 

---

632
ent clustering algorithms and objective functions through various notions of the NCP plot and other kinds of structural metrics of clusters and how they depend/scale with the size of the cluster.

Moreover, the shape of the NCP is also interesting for a very different reason. It gives us a powerful way to quantify and summarize the large-scale community structure of networks. We [27, 26] found that the NCP behaves in a characteristic manner for a range of large social and information networks: when plotted on log-log scales, the NCP tends to have a universal "V" shape (Figure 1(right)). Up to a size scale of about 100 nodes, the NCP decreases, which means that the best-possible clusters are getting progressively better with the increasing size. The NCP then reaches the minimum at around $k = 100$ and then gradually increases again, which means that at larger size scales network communities become less and less community-like. (This should be contrasted with behavior for mesh-like networks, road networks, common network generation models, and small commonly-studied networks, for which the NCP is either flat or downward-sloping [27, 26].)

The shape of the NCP can be explained by an onion-like “nested core-periphery” structure, where the network consists of a large core (slightly denser and more expander-like than the full graph, but which itself has a core-periphery structure) and a large number of small very well-connected communities barely connected to the core [27, 26]. In this context, it is important to understand the characteristics of various community detection algorithms in order to make sure that the shape of NCP is a property of the network rather than an artifact of the approximation algorithm or the function that formalizes the notion of a network community.

3. COMPARISON OF ALGORITHMS

We compare different clustering algorithms and heuristics. We focus our analyses on two aspects. First, we are interested in the quality of the clusters that various methods are able to find. Basically, we would like to understand how well algorithms perform in terms of optimizing the notion of community quality (conductance in this case). Second, we are interested in quantifying the structural properties of the clusters identified by the algorithms. As we will see, there are fundamental tradeoffs in network community detection—for a given objective function, approximation algorithms are often biased in a sense that they consistently find clusters with particular internal structure.

We break the experiments into two parts. First, we compare two graph partitioning algorithms that are theoretically well understood and are based on two very different approaches: a spectral-based Local Spectral partitioning algorithm, and the flow-based Metis+MQI. Then we consider several heuristic approaches to network community detection that work well in practice.

3.1 Flow and spectral methods

In this section we compare the Local Spectral Partitioning algorithm [2] with the flow-based Metis+MQI algorithm. The latter is a surprisingly effective heuristic method for finding low-conductance cuts, which consists of first using the fast graph bi-partitioning program Metis [20] to split the graph into two equal-sized pieces, and then running MQI, an exact flow-based technique [15, 23] for finding the lowest conductance cut whose small side in contained in one of the two half-graphs chosen by Metis.

Each of those two methods (Local Spectral and Metis+MQI) was run repeatedly with randomization on each of our graphs, to produce a large collection of candidate clusters of various sizes, plus a lower-envelope curve. The lower-envelope curves for the two algorithms were the basis for the plotted NCP’s in the earlier paper [27]. In the current paper the lower-envelope curves for Local Spectral and Metis+MQI are plotted respectively as a red line and a green line in Figure 1(right), and as pairs of black lines in Figure 2(top) and Figures 4 and 7. Note that the Metis+MQI curves are generally lower, indicating that this method is generally better than Local Spectral at the nominal task of finding cuts with low conductance.

However, as we will demonstrate using the scatter plots of Figure 2, the clusters found by the Local Spectral Method often have other virtues that compensate for their worse conductance scores. As an extreme example, many of the raw Metis+MQI clusters are internally disconnected, which seems like a very bad property for an alleged community. By contrast, the Local Spectral Method always returns connected clusters. Acknowledging that this is a big advantage for Local Spectral, we then modified the collections of raw Metis+MQI clusters by splitting every internally disconnected cluster into its various connected components. Then, in all scatter plots of Figure 2, blue dots represent raw Local Spectral clusters, which are internally connected, while red dots represent broken-up Metis+MQI clusters, which are also internally connected.

Let us now consider the top row of scatter plots of Figure 2 which compares the conductance scores (as a function of cluster size) of the collections of clusters produced by the two algorithms. The cloud of blue points (Local Spectral clusters) lies generally above the cloud of red points (Metis+MQI clusters), again illustrating that Local Spectral tends to be a weaker method for minimizing conductance score. In more detail, we find that Local Spectral and Metis+MQI tend to identify similar pieces at very small scales, but at slightly larger scales a gap opens up between the red cloud and the blue cloud. At those intermediate size scales, Metis+MQI is finding lower conductance cuts than Local Spectral.

However, the Local Spectral algorithm returns pieces that are internally more compact. This is shown in the middle row of Figure 2 where for each of the (connected) pieces for which we plotted a conductance in the top row, we are now plotting the average shortest path length between random node pairs in that piece. In these plots, we see that in the same size range where Metis+MQI is generating clearly lower conductance connected sets, Local Spectral is generating pieces with clearly shorter internal paths, i.e., smaller diameter sets. In other words, the Local Spectral pieces are more "compact." This effect is especially pronounced in the DBLP affili-
Conductance of connected clusters found by Local Spectral (blue) and Metis+MQI (red)

Cluster compactness: average shortest path length

Cluster compactness: external vs. internal conductance

Figure 2: Comparison of Local Spectral (blue) and Metis+MQI (red) on connected clusters. Top: the conductance of the bounding cut. Middle: the average shortest path length in the cluster. Bottom: the ratio of the external conductance to the internal conductance. Generally Metis+MQI yields better cuts while Local Spectral yields clusters that are more compact: they have shorter path lengths and better internal connectivity.

...ation network, while it also shows up in the Enron email network and the astrophysics collaboration network. Moreover, we made similar observations also for many other datasets (plots not shown).

Finally, in the bottom row of Figure 2 we introduce the topic of internal vs. external cuts, which is something that none of the existing algorithms is explicitly optimizing. These are again scatter plots showing the same set of Local Spectral and Metis+MQI pieces as before, but now the y-axis is external conductance divided by internal conductance. External conductance is the quantity that we usually plot, namely the conductance of the cut which separates the cluster from the graph. Internal conductance is the score of a low conductance cut inside the cluster. That is, we take the induced subgraph on the cluster’s nodes and then find best conductance cut inside the cluster.

We then compare the ratios of the conductance of the bounding cut and the internal conductance. Intuitively, good and compact communities should have small ratios, ideally below 1.0, which would mean that those clusters are well separated from the rest of the network and that they are also internally well-connected and hard to cut again. However, the three bottom-row plots of Figure 2 show the ratios. Points above the horizontal line are clusters which are easier to cut internally than they were to be cut from the rest of the network; while points below the line are clusters that were relatively easy to cut from the network and are internally well-connected. Notice that here the distinction between the two methods is less clear. On the one hand, Local Spectral finds clusters that have worse (higher) bounding cut conductance, while such clusters are also internally more compact (have internal cuts of higher conductance). On the other hand, Metis+MQI finds clusters that have better (lower) bounding cut conductance but are also internally easy to cut (have internal cut of lower conductance). Thus when one takes the ratio of the two quantities we observe qualitatively similar behaviors. However, notice that Local Spectral seem to return clusters with higher variance in the ratio of external-to-internal conductance. At small size scales Metis+MQI tends to give clusters of slightly better (lower) ratio, while at larger clusters the advantage goes to Local Spectral. This has interesting consequence for the applications of graph partitioning since (depending on the particu-
lar application domain and the sizes and properties of clusters one aims to extract) either Local Spectral or Metis+MQI may be the method of choice.

Also, notice that there are mostly no ratios well below 1.0, except for very small sizes. This is important, as it seems to hint that large clusters are relatively hard to cut from the network, but are then internally easy to split into multiple sub-clusters. This shows another aspect of our findings: small communities below \( \approx 100 \) nodes are internally compact and well separated from the remainder of the network, whereas larger clusters are so hard to separate that cutting them from the network is more expensive than cutting them internally. Community-like sets of nodes that are better connected internally than externally don’t seem to exist in large real-world networks, except at very small size scales.

Last, in Figure 3, we further illustrate the differences between spectral and flow-based clusters by drawing some example subgraphs. The two subgraphs shown on the left of Figure 3 were found by Local Spectral, while the two subgraphs shown on the right of Figure 3 were found by Metis+MQI. These two pairs of subgraphs have a qualitatively different appearance: Metis+MQI pieces look longer and stringier than the Local Spectral pieces. All of these subgraphs contain roughly 500 nodes, which is about the size scale where the differences between the algorithms start to show up. In these cases, Local Spectral has grown a cluster out a bit past its natural boundaries (thus the spokes), while Metis+MQI has strung together a couple of different sparsely connected clusters. (We remark that the tendency of Local Spectral to trade off cut quality in favor of piece compactness isn't just an empirical observation, it is a well understood consequence of the theoretical analysis of spectral partitioning methods.)

3.2 Other algorithms

Next we consider various other, mostly heuristic, algorithms and compare their performance in extracting clusters of various sizes. As a point of reference we use results obtained by the Local Spectral and Metis+MQI algorithms.

We have extensively experimented with several variants of the global spectral method, both the usual eigenvector-based embedding on a line, and an SDP-based embedding on a hypersphere, both with the usual hyperplane-sweep rounding method and a flow-based rounding method which includes MQI as the last step. In addition, special post-processing can be done to obtain either connected or disconnected sets.

We also experimented with a practical version of the Leighton-Rao algorithm [24, 25], similar to the implementation described in [22, 23]. These results are especially interesting because the Leighton-Rao algorithm, which is based on multi-commodity flow, provides a completely independent check on Metis, and on spectral methods generally. The Leighton-Rao algorithm has two phases. In the first phase, edge congestions are produced by routing a large number of commodities through the network. We adapted our program to optimize conductance (rather than ordinary ratio cut score) by letting the expected demand between a pair of nodes be proportional to the product of their degrees. In the second phase, a rounding algorithm is used to convert edge congestions into actual cuts. Our method was to sweep over node orderings produced by running Prim’s Minimum Spanning Tree algorithm on the congestion graph, starting from a large number of different initial nodes, using a range of different scales to avoid quadratic run time. We used two variations of the method, one that produces connected sets, and another one that can also produce disconnected sets.

In top row of Figure 4, we show Leighton-Rao curves for three example graphs. Local Spectral and Metis+MQI curves are drawn in black, while the Leighton-Rao curves for connected and possibly disconnected sets are drawn in green and magenta respectively. For small to medium scales, the Leighton-Rao curves for connected sets resemble the Local Spectral curves, while the Leighton-Rao curves for possibly disconnected sets resemble Metis+MQI curves. This further confirms the structure of clusters produced by Local Spectral and Metis+MQI, as discussed in Section 3.1.

At large scales, the Leighton-Rao curves shoot up and become much worse than Local Spectral or Metis+MQI. That Leighton-Rao has trouble finding good big clusters is not surprising because expander graphs are known to be the worst case input for the Leighton-Rao approximation guarantee. Large real networks contain an expander-like core which is necessarily encountered at large scales. We remark that Leighton-Rao does not work poorly at large scales on every kind of graph. (In fact, for large low-dimensional mesh-like graphs, Leighton-Rao is a very cheap and effective method for finding cuts at all scales, while our local spectral method becomes impractically slow at medium to large scales.) This means that based on the structure of the network and sizes of clusters one is interested in different graph partitioning methods should be used. While Leighton-Rao is an appropriate method for mesh-like graphs, it has troubles in the intermingled expander-like core of large networks.

Finally, in addition to the above approximation algorithms-based methods for finding low-conductance cuts, we also experimented with a number of more heuristic approaches that tend to work well in practice. In particular, we compare Graclus [9] and Newman’s modularity optimizing program (we refer to it as Dendrogram) [16]. Graclus attempts to partition a graph into pieces bounded by low-conductance cuts using a kernel \( \mathcal{K} \)-means algorithm. We ran Graclus repeatedly, asking for \( 2, 3, \ldots, \lfloor \sqrt{2} \rfloor \) pieces. Then we measured the size and conductance of all of the resulting pieces. Newman’s Dendrogram algorithm constructs a recursive partitioning of a graph (that is, a dendrogram) from the bottom up by repeatedly deleting the surviving edge with the highest betweenness centrality. A flat partitioning could then be obtained by cutting at the level which gives the highest modularity score, but instead of doing that, we measured the size of conductance of every piece defined by a subtree in the dendrogram.

The bottom row of Figure 4 presents these results. Again our two standard curves are drawn in black. The lower-envelopes of the Graclus or Dendrogram points are roughly similar to those produced by Local Spectral, which means both methods tend to produce rather compact clusters at all size scales. Generally, Graclus tends to produce a variety of clusters of better conductance than Newman’s algorithm. Moreover, notice that in case of Epinions social network and the astrophysics coauthorship network Graclus tends to prefer larger clusters than the Newman’s algorithm. Also, Graclus seems to find clusters of ten or more nodes, while Newman’s algorithm also extracts very small pieces. In general, clusters produced by either Graclus or Dendrogram are qualitatively similar to those produced by Local Spectral. This means that even though Local Spectral is computationally cheaper and easily scales to very large networks, the quality of identified clusters is comparable to that returned by techniques such as Graclus and Dendrogram that are significantly more expensive on large networks such as those we considered.

4. COMPARISON OF OBJECTIVE FUNCTIONS

In the previous sections, we used conductance since it corresponds most closely to the intuition that a community is a set of
nodes that is more and/or better connected internally than externally. In this section, we look at other objective functions that capture this intuition and/or are popular in the community detection literature.

In general there are two criteria of interest when thinking about how good of a cluster is a set of nodes. The first is the number of edges between the members of the cluster, and the second is the number of edges between the members of the cluster and the remainder of the network. We group objective functions into two groups. The first group, that we refer to as Multi-criterion scores, combines both criteria (number of edges inside and the number of edges crossing) into a single objective function; while the second group of objective functions employs only a single of the two criteria (e.g., volume of the cluster or the number of edges cut).

### 4.1 Multi-criterion scores

Let $G(V, E)$ be an undirected graph with $n = |V|$ nodes and $m = |E|$ edges. Let $S$ be the set of nodes in the cluster, where $n_S$ is the number of nodes in $S$, $n_S = |S|$; $m_S$ the number of edges in $S$, $m_S = \{(u, v) : u \in S, v \in S\}$; and $c_S$, the number of edges on the boundary of $S$, $c_S = \{(u, v) : u \in S, v \notin S\}$; and $d(u)$ is the degree of node $u$.

We consider the following metrics $f(S)$ that capture the notion of a quality of the cluster. Lower value of score $f(S)$ (when $|S|$ is kept constant) signifies a more community-like set of nodes.

- **Conductance**: $f(S) = \frac{c_S}{2m_S}$ measures the fraction of total edge volume that points outside the cluster [32, 18].
- **Expansion**: $f(S) = \frac{c_S}{d(u)}$ measures the number of edges per node that point outside the cluster [30].

---

Figure 3: Two examples of clusters found by the LocalSpectral algorithm (on the left) and two from the Metis+MQI algorithm (on the right). Note that the Local Spectral clusters are more compact—they are tighter and have smaller diameter since the algorithm has difficulty pushing probability mass down long extended paths—while the Metis+MQI clusters are more sprawling—they have larger diameter and more diverse internal structure, but better conductance scores. In both cases, we have shown communities with ca. 500 nodes (many of nodes overlap at resolution of this figure).

Figure 4: Comparison of various algorithms on Epinions, Email-Enron, and CA-Astro-Ph. Top row: NCP plots for connected (green) and disconnected (magenta) pieces from our implementation of the Leighton-Rao algorithm. Bottom row: conductance of cuts found by Graclus and by Newman’s Dendrogram algorithm. Notice the qualitative shape of the NCP plots remains practically unchanged regardless of what particular community detection algorithm we use. Small clusters tend to be connected, while for clusters larger than about 100 nodes connected clusters have around 5 times worse conductance score than disconnected clusters.
Internal density: $f(S) = 1 - \frac{m_S}{n_S(n_S-1)/2}$ is the internal edge density of the cluster $S$ [30].

Cut Ratio: $f(S) = \frac{c_S}{n_S(n_S-1)/2}$ is the fraction of all possible edges leaving the cluster [12].

Normalized Cut: $f(S) = \frac{c_S}{2m_S} + \frac{c_g}{2(m_S-n_S)+c_g}$ [32].

Maximum-ODF (Out Degree Fraction): $\max_{u \in S} \frac{d(u) - c_S}{|S|}$ is the maximum fraction of edges of a node pointing outside the cluster [10].

Average-ODF: $f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{u,v \in S : d(u) < d(v)/2 \}|}{d(u)}$ is the average fraction of nodes’ edges pointing outside the cluster [10].

Flake-ODF: $f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{u,v \in S : d(u) < d(v)/2 \}|}{d(u)/2}$ is the fraction of nodes in $S$ that have fewer edges pointing inside than to the outside of the cluster [10].

We then generalize the NCP plot: for every cluster size $k$ we find a set of nodes $S$ \(|S| = k\) that optimizes the chosen community score $f(S)$. We then plot community score as a function of $k$. It is not clear how to design an optimization procedure that would, given a cluster size $k$ and the community score function $f(S)$, find the set $S$ that minimizes the function, i.e., is the best community. Operationally, we perform the optimization the following way: we use the Local Spectral method which starts from a seed node and then explores the cluster structure around the seed node; running Local Spectral from each node, we obtain a millions of sets of nodes of various sizes, many of which are overlapping; and then for each such set of nodes, we compute the community score $f(S)$ and find the best cluster of each size.

Figure 5 considers the above eight community scores. Notice that even though scores span different ranges they all experience qualitatively similar behavior, where clusters up to size ca. 100 have progressively better scores, while the clusters above ca. 100 nodes become less community-like as their size increases. This may seem surprising at the first sight, but it should be somewhat expected, as all these objective functions try to capture the same basic intuition—they reward sets of nodes that have many edges internally and few pointing out of the clusters.

There are, however, subtle differences between various scores. For example, even though Flake-ODF follows some general trend as conductance, it reaches the minimum about an order of magnitude later than conductance, normalized cut, cut ratio score or the Average-ODF. The other hand, Maximum-ODF exhibits the opposite behavior as it clearly prefers smaller clusters and is basically flat for clusters larger than about several hundred nodes. This is interesting as this shows the following trend: if one scores the community by the “worst-case” node using the Out Degree Fraction (i.e., Maximum-ODF) then only small clusters have no outliers and thus give good scores. When one considers the average fraction of node’s edges pointing outside the cluster (Average-ODF) the objective function closely follows the trend of conductance. On the other hand, if one considers the fraction of nodes in the cluster with more of their edges pointing inside than outside the cluster (Flake-ODF), then large clusters are preferred.

Next, focusing on the cut ratio score we notice that it is not very smooth, in the sense that even for large clusters its values seem to fluctuate quite a lot. This indicates that clusters of similar sizes can have very different numbers of edges pointing to the rest of the network. In terms of their internal density, the variations are very small—the internal density reaches the maximum for clusters of sizes around 10 nodes and then quickly raises to 1, which means larger clusters get progressively sparser. For large clusters this is not particularly surprising as the normalization factor increases quadratically with the cluster size. This can be contrasted with the Expansion score that measures the number of edges pointing outside the cluster but normalizes by the number of nodes (not the number of all possible edges).

These experiments suggest that Internal Density and Maximum-ODF are not particularly good measures of community score and the cut ratio score may not be preferred due to high variance. Flake-ODF seems to prefer larger clusters, while conductance, expansion, normalized cut, and Average-ODF all exhibit qualitatively similar behaviors and give best scores to similar clusters.

In addition, we performed an experiment where we extracted clusters based on their conductance score but then also computed the values of other community scores (for these same clusters). This way we did not optimize each community score separately, but rather we optimized conductance and then computed values of other objective functions on these best-conductance pieces. The shape of the plots remained basically unchanged, which suggests that same sets of nodes achieve relatively similar scores regardless of which particular notion of community score is used (conductance, expansion, normalized cut, or Average-ODF). This shows that these four community scores are highly correlated and in practice prefer practically the same clusters.

### 4.2 Single criterion scores

Next we also consider community scores that consider a single criteria. One such example is Modularity [28], which is one of the most widely used methods to evaluate the quality of a division of a network into modules or communities. For a given partition of a network into clusters, modularity measures the number of within-community edges, relative to a null model of a random graph with the same degree distribution.

Here we consider the following four notions of a quality of the community that are based on using one or the other of the two criteria of the previous subsection:

- **Modularity**: $\frac{1}{m} (m_S - E(m_S))$, where $E(m_S)$ is the expected number of edges between the nodes in set $S$ in a random graph with the same node degree sequence.

- **Modularity ratio**: $\frac{m_S}{E(m_S)}$ is alternative definition of the modularity, where we take the ratio of the number of edges between the nodes of $S$ and the expected number of such edges under the null-model.

- **Volume**: $\sum_{u \in S} d(u)$ is sum of degrees of nodes in $S$.

- **Edges cut**: $c_S$ is number of edges needed to be removed to disconnect nodes in $S$ from the rest of the network.

Figure 6 shows the analog of the NCP plot where now instead of conductance we use these four measures. A general observation is that modularity tends to increase roughly monotonically towards the bisection of the network. This should not be surprising since modularity measures the “volume” of communities, with (empirically, for large real-world networks) a small additive correction, and the volume clearly increases with community size. On the other hand, the modularity ratio tends to decrease towards the bisection of the network. This too should not be surprising, since it involves dividing the volume by a relatively small number. Results in Figure 6 demonstrate that, with respect to the modularity, the “best” community in any of these networks has about half of all nodes; while, with respect to the modularity ratio, the “best” community in any of these networks has two or three nodes.

Leaving aside debates about community-quality objective functions, note that, whereas the conductance and related measures are discriminative, in that they prefer different kinds of clusters, depending on the type of network being considered, modularity tends
Figure 7: Lower bounds on best conductance cut. In black we plot conductance curves as obtained by Local Spectral and Metis+MQI. Lower bounds on conductance of any cut (Spectral lower bound, dashed line) and the cut separating the graph in half (SDP lower bound, red triangle) show that both algorithms do a good job on finding best possible and best balanced cut of the network.


