IsoClustering: A Generalized Framework for Local Data Clustering

Abstract—**In this paper, we propose a generalized framework for local clustering based on isoperimetric inequalities. We also demonstrate that contemporary approaches are included in its scope and that it can accommodate data sets of different types, including those with overlapping communities. We then present an efficient, greedy algorithm using the new framework and compare the output of the new algorithm with existing methods.**

Keywords*—*data clustering; community detection; local clustering; overlapping communities; isoperimetric inequality

I. Introduction

One of the most fundamental problems in analyzing big data sets is the task of data clustering. In a large number of practical applications, data can be represented as graphs in which vertices represent the objects in the data set and weighted edges between the vertices represent the pairwise similarity of the two objects. The data clustering problem is to identify groups of vertices that are strongly connected within their groups but weakly connected to the vertices outside their respective groups. These groups are referred to as clusters, and the goal in finding such clusters is to make clear the underlying structure of the data. For example, the task of classifying web pages into topically similar groups can be resolved as a clustering problem.

Because of its usefulness in a broad range of scientific applications, many methods have been developed to address the problem of clustering, and it remains a very active area of research. A myriad of methods have been introduced to partition graphs into clusters, optimize global metrics like modularity, and hierarchically merge vertices, among other approaches. These and other popular methods are well summarized in a work by Fortunato [1]. While the great majority of this work applies to unweighted graphs (i.e. the Community Detection problem), in most cases these methods can be extended to weighted graphs.

While much progress on the clustering problem has been made over the past several decades, a perennial issue in pursuing such research is that on a fundamental level the problem is ill-posed, as a precise definition of cluster is still missing. The result is that research is scattered, as each researcher interprets the goal of the clustering problem differently. That said, some attempts have been made to place bounds and restrictions on what the properties of an acceptable clustering of a data set might be [2]–[5]. For example, Lu et al detail the concept of "minimum community conditions" [2]. However, this establishes only a lower bound on what an acceptable cluster might look like, and in practice communities that meet this minimum condition may be suspect in quality. As we will discuss in a later section, even complete graphs have proper subgraphs that locally satisfy the community conditions, which defies the intuitive sense of a cluster.

A particular difficulty encountered while trying to classify the clusters in a data set is that methods that rely on partitioning do not include the possibility that clusters overlap. In social networks, for instance, one person may belong to several different circles of friends. Trying to partition the data in a way that assigns such an individual to only one cluster leads to an obfuscation of the true phenomenon being described. Thus, rather than partition the data, it is useful instead to find a cover – that is, a collection of overlapping clusters that includes at least one assignment for each vertex [1]. Finding a cover is a more difficult task than simply partitioning the data, and so to combat this difficulty encountered when clusters overlap, it is useful to have a local description of clusters. By clustering the data according to a local definition, the inclusion of a node in one cluster does not preclude its presence in another since there is no global assignment. Additionally, a local definition of cluster allows for solutions in cases where the entire data set is not known, as is often the case in dynamic clustering [6].

In this paper we propose a framework for the local definition of clusters that is general enough to describe existing clustering methods, but precise enough to greatly narrow the range of possible algorithms. Additionally, we connect the clustering problem with a well-developed area of mathematical theory. We borrow the concept of isoperimetric inequalities on general metric spaces [7]. Then we identify it with clustering in a way that from any particular definition of volume and consistent definition of perimeter, a precise local clustering criterion is induced. It is then sufficient to assert what measure of volume and perimeter is appropriate for the nature of the data being analyzed, for then the definition of clusters in that particular data set will become apparent.

II. Theoretical Preliminaries

A. Isoperimetric Inequalities on Graphs

To motivate our approach it is first necessary to illustrate the connection between the clusters of a graph and isoperimetric inequalities. Recall that two clusters on a graph should be highly connected within and loosely connected to each other. This situation can be seen analogously as a bottleneck for the topology of the graph, hence one can expect the bottlenecks to represent the boundaries of the clusters.

To clarify the relation between a bottleneck and a minimizer of $\frac{P}{V}$ (i.e. the isoperimetric minimizer), consider a simple space made of the surface of a dumbbell, as in Fig. 1. For any simple closed curve on this surface, the 2-dimensional volume is the area within the closed curve and the perimeter is the usual length of the curve. Compare two typical closed

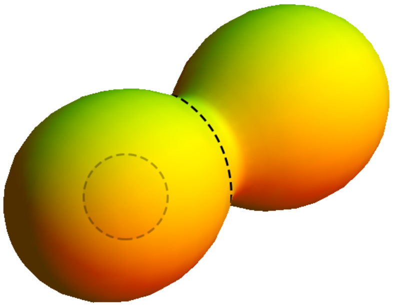

Fig. 1. A simple manifold demonstrating that the minimizer of the isoperimetric inequality is at the bottleneck.

curves on the surface, one being the dotted circle on the left handle and the other curve being the dotted curve close to the bottleneck of the dumbbell. It is visually clear that while the length is not significantly different, the area enclosed is much bigger for the curve at the bottleneck. More generally, given a Riemannian manifold, the isoperimetric inequality is an inequality governing the ratio of the volume of a shape in this space to its perimeter, with the optimizers of the inequality being the bottlenecks of the space. In such a manner, we are interested in finding the bottlenecks of a graph, and in such a case one can discretize the isoperimetric minimizer with any reasonable definition of volume and consistent definition of perimeter.

There are many different possible ways to define the volume of a subgraph, with common methods being to use the sum of the edge weights, the sum of the degree sequence, or simply the amount of vertices present within the cluster. Declaring a particular definition for volume induces a family of consistent perimeter definitions that are given by

$$P(C) = \frac{V(C_\epsilon \cup C) - V(C)}{\epsilon}$$

with different variations possible depending upon how C_ϵ is interpreted in context. For instance, if volume is defined as the number of vertices within a cluster, then one possibility for the definition of perimeter could be the number of vertices that can be reached from vertices in the cluster via paths of length ϵ. This notion of ϵ penetration can be modified should the nature of the data suggest a different interpretation, but the end result is that the range of perimeter definitions that are consistent with the volume definition will be lesser in scope than if the perimeter definition was arbitrary.

Given consistent definitions for volume and perimeter, we can attempt to find clusters that locally minimize the value of $\frac{P}{V}$. In practice, it is desirable to find the clusters that *most locally* minimize this value since local minima can correspond to unions of smaller clusters and the global minimum for a connected graph will almost certainly occur when the entire graph is considered as one cluster. We will refer to the process of finding these most local clusters as **IsoClustering** to distinguish the results from the larger family of covers that can be constructed from their unions. As we are looking for a very local feature of the data, greedy algorithms such as the one that we present in our paper have a particular advantage in that they are not only an efficient way to find such minima, but they also are more likely to find local phenomena.

As the minimization problem is presented here in abstract form, one is not limited to any particular metric, but instead limited only by the initial definition of volume and a consistent definition of perimeter. As we will see in a following section, particular choices of such definitions induce many existing algorithms, yet the overall scope still leaves open the possibility for future exploration and provides a direction for such exploration to occur.

B. IsoClustering versus Minimum Community Conditions

An early attempt to provide a framework for cluster definitions involved the introduction of "minimum community conditions" [2]. There are many such conditions, ranging from weak to strong, with the weakest definition being defined as follows:

Definition 1: Given a connected and undirected simple graph $G(V, E)$, let W denote the adjacency matrix. A partition on vertices $\mathcal{P} = \{C_1, C_2, \cdots, C_K\}$ with $\bigcup_{k=1}^{K} C_k = V$ and $C_k \cap C_t = \emptyset$ $(k \neq t)$, is called a valid community set of G if $\forall 1 \leq k \leq K$ the sub-graph $G[C_k]$ is connected and satisfies $\sum_{\forall i,\ j \in C_k} A_{i,j} > \sum_{\forall i \in C_k,\ \forall j \in C_t} A_{i,j},\ \forall k \neq t$.

Originally envisioned as a means of defining clusters via partitioning, the usefulness of such minimal conditions for local clustering is suspect, since complete graphs with at least five vertices have proper subgraphs that satisfy the final inequality. For example, in the complete graph on five vertices, a cluster containing any four nodes has 6 interior edges but only 4 exterior edges. By contrast, IsoClustering will produce only one cluster in a complete graph (by producing the cluster consisting of the entire graph), thus correctly identifying the lack of community structure in complete graphs of all sizes.

C. Stopping Criterion and Resolution

As $d\left(\frac{P}{V}\right)$ can be written as $d\left(\frac{P}{V}\right) = \frac{P}{V}\left(\frac{dP}{P} - \frac{dV}{V}\right)$, for $P, V > 0$, the resulting optimality criterion is $\frac{dP}{P} > \frac{dV}{V}$ over all possible propagation vectors. By interpolating between these two terms, we can insert a resolution parameter $\alpha \in [0, 1]$, by which we can control the strictness with which the inequality is enforced, allowing for more or less strict enforcement of bottlenecks. This parameter then serves to bypass the so-called resolution limit that is the drawback of many modularity-based algorithms [8]. As a result, the stopping criterion then becomes $\alpha\left(\frac{dP}{P}\right) > (1-\alpha)\left(\frac{dV}{V}\right)$ over all possible propagation vectors.

By variation of the resolution parameter, we can control the number and size of clusters found. Such a parameter is desirable since in the absence of a ground truth regarding the number of clusters to find, a local algorithm cannot determine the amount of noise in the data; that is, it cannot distinguish between sampling error and the actual features of the data. Thus, through the choice of α we are able to impart some sense of signal-to-noise ratio, and as we will see later on, through constructive choice of this parameter we achieve some very desirable results.

III. Algorithm

A. Algorithm Summary

1) Start a new cluster that contains only $starting\_node$.

Algorithm 1 IsoClustering Algorithm

```
Input: $G$, $W$, $\alpha$, $starting\_node$;
$C = \emptyset$;
$x = starting\_node$;
do
    $C = C \cup x$;
    ratios[j] = $\frac{dV[j]}{dP[j]}$;        ▷ calculate for all j $\notin$ C
    $x = \arg\max$(ratios);
while ($C \neq G$ and $\alpha \cdot dP[x] \cdot V \leq (1-\alpha) \cdot dV[x] \cdot P$);
```

2) Find the unclassified node with the highest value of weight into the developing cluster divided by weight into its complement.
3) If bringing the new node into the cluster will satisfy the inequality, bring this node into the cluster, and then repeat 2) - 3) for as long as the inequality is satisfied.

This algorithm then produces one cluster containing the starting node.

As the algorithm finds each cluster independently based upon a particular starting node, the classification of a node in one cluster does not influence the decision to include it in other clusters. Repeating the process by using multiple starting nodes can produce many such clusters, and any overlap will be realized as the inclusion of a node in more than one unique cluster.

Another intended benefit of the independence of each pass is that it is possible (and likely, desirable) to conduct the process in parallel. By assigning to each processor a separate starting node and removing duplicates in a merge step, a thorough clustering can be found.

If a complete partitioning of the graph should be desired, after each iteration the algorithm can be run again on the subgraph consisting only of vertices that have not already been classified in a prior iteration. As with all partitioning algorithms, results will vary in the case that the underlying structure of the graph contains clusters with a high degree of overlap.

IV. Special Cases: L^p Volume

While we have in our presentation of the topic intentionally left particular definitions such as volume generic, we feel that there is a special case worth calling out specifically, as it demonstrates the connection of these ideas to existing clustering theory. Consider defining volume as

$$V = \sqrt[p]{\sum_{i,j \in C} W_{ij}^p} \ \forall p > 0.$$

This induces a family of consistent perimeter formulas as $P(C) = \frac{V(C_\epsilon \cup C) - V(C)}{\epsilon}$. Let us interpret C_ϵ to mean the union of edges with both endpoints in the cluster with the edges that have only one endpoint in the cluster. Here we choose units for ϵ so that it has a value of 1 in order to simplify the next calculation, and as a result, use

$$P = \sqrt[p]{\sum_{i \in C, j \notin C} W_{ij}^p + \sum_{i,j \in C} W_{ij}^p} - \sqrt[p]{\sum_{i,j \in C} W_{ij}^p}$$

as the definition of perimeter for the remainder of this section.

Then,

$$\begin{aligned}
&\arg\min_C \left(\frac{P}{V}\right) \\
&= \arg\min_C \left(\frac{\sqrt[p]{\sum_{i \in C, j \notin C} W_{ij}^p + \sum_{i,j \in C} W_{ij}^p} - \sqrt[p]{\sum_{i,j \in C} W_{ij}^p}}{\sqrt[p]{\sum_{i,j \in C} W_{ij}^p}} \right) \\
&= \arg\min_C \left(\sqrt[p]{1 + \frac{\sum_{i \in C, j \notin C} W_{ij}^p}{\sum_{i,j \in C} W_{ij}^p}} - 1 \right) \\
&= \arg\min_C \left(\sqrt[p]{1 + \frac{\sum_{i \in C, j \notin C} W_{ij}^p}{\sum_{i,j \in C} W_{ij}^p}} \right) \\
&= \arg\min_C \left(1 + \frac{\sum_{i \in C, j \notin C} W_{ij}^p}{\sum_{i,j \in C} W_{ij}^p} \right) \\
&= \arg\min_C \left(\frac{\sum_{i \in C, j \notin C} W_{ij}^p}{\sum_{i,j \in C} W_{ij}^p} \right).
\end{aligned}$$

Then minimizing $\frac{P}{V}$ based on these definitions is equivalent to minimizing

$$\frac{\sum_{i \in C, j \notin C} W_{ij}^p}{\sum_{i,j \in C} W_{ij}^p}.$$

Combining this minimization with the aforementioned algorithm and efficient bookkeeping, it is possible to implement this approach with $O(mn)$ complexity, where n is the number of nodes present in the graph and m is the number of classifications that will be made. If successive iterations are performed only on the complement of the graph (i.e. the partitioning variation of the algorithm), it can produce a complete clustering as $O(n^2)$.

That this approach is a productive one has in part been already explored. In the particular case that $p = 1$, we have the result corresponding to the L^1 norm, namely to minimize

$$\frac{\sum_{i \in C, j \notin C} W_{ij}}{\sum_{i,j \in C} W_{ij}}.$$

In literature this is often referred to as the *Cheeger Cut* and has been well-explored by modern research [10]. That the choice of $p = 2$ is also productive has been well-known for some time, as such a choice allows for the mathematics of Hilbert spaces to be used, and so we will not detail the plethora of spectral methods here (see [1]).

What we attempt to bring to the forefront by calling out this special case of L^p volumes is that we do not limit ourselves to the typical choices of $p = 1$ and $p = 2$, which so often form

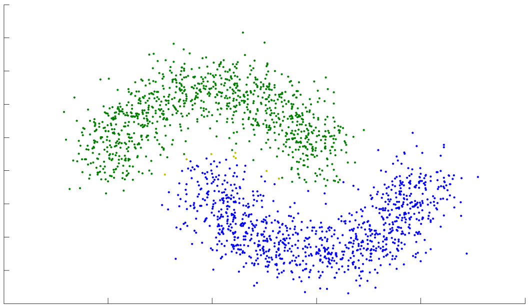

Fig. 2. Clustering of two moons using α = 0.45, p = 0.5.

the basis of contemporary algorithms, and instead have found that some very useful results come from choices of $p < 1$. That such choices can produce competitive results could prove to be a valuable area of future research, and we will show some specific examples in the next section.

V. Results

Tests on synthetic data with many data points ($n > 1000$) show that the algorithm produces good separation of clusters and meaningful overlap. Fig. 2 is a representative example. The data set is sampled from two moons in close proximity, and the graph is constructed by applying a Gaussian kernel to the distance between points, normalized to the 50^{th} nearest neighbor but not sparsified. The detected clusters are shown in blue and green, and the overlapping section in yellow. Note that the algorithm not only produces excellent purity on this data set, but also correctly identifies that there are two clusters.

In practice, the number of clusters varies naturally with the choice of the parameters α and p, and so the choice of such parameters is non-trivial. Consider Fig. 3, which shows how the choice of α can affect the minimum number of clusters that cover the entire data set. We have shown the resolution analysis on the Zachary Karate Club data set, as it demonstrates that our algorithm can reproduce the results of similar published analysis [9]. By its construction, α interpolates between the relative change in perimeter and the relative change in volume. For $\alpha = 0.5$, each is weighted equally – the same as if no resolution parameter had been used. If α is chosen below 0.5, the algorithm can proceed past weak bottlenecks, thus finding fewer clusters. If α is chosen above 0.5, the algorithm enforces bottlenecks more strictly, and thus more clusters are found.

The qualitative effect of α can be better seen in Fig. 4, which shows the effect of relaxing α for a data set constructed of two radial Gaussians. Again, the two clusters are shown in blue and green, and their overlap in yellow. α has been relaxed so that the algorithm is not as strict in enforcing the stopping condition, and the result is that points which are not closely tied to either center are included in the overlap.

By contrast, Fig. 5 shows the effect of relaxing p. Here the algorithm produces different results – there is less overlap between the clusters and the geometric decision boundary is more evident. Curiously, the best results occur for values of p less than one, and in our experiments the results are generally best when p is chosen in the range of $(0.4, 0.6)$.

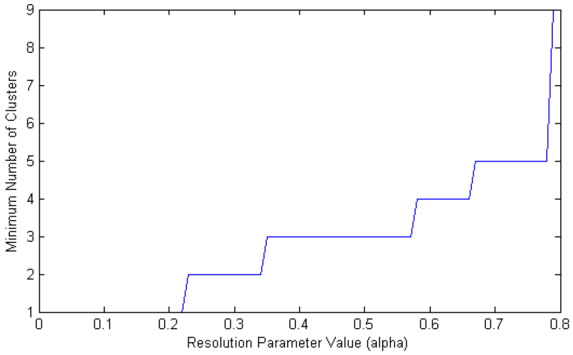

Fig. 3. Minimum Cluster Formation versus α for Zachary Karate Dataset.

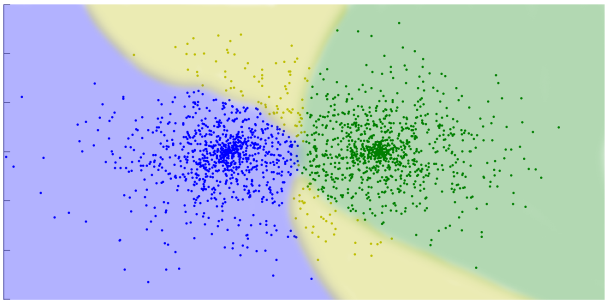

Fig. 4. Clustering of two Gaussians using α = 0.4, p = 1.0.

Note that in each case we have not supplied the algorithm with the number of clusters it should find, but instead choose an appropriate resolution and L^p volume.

Fig. 6, 7, and 8 serve to illustrate the robust quality of the algorithm. Here the data set is sampled from two interlocking rings in three dimensions that do not touch. The graph is constructed with a Gaussian kernel as described before. IsoClustering correctly identifies both clusters and their lack of overlap, while spectral clustering is unable to correctly classify this type of data set at all. In fact, the IsoClustering results match perfectly with that of the Cheeger Cut [10], but involve significantly less computational complexity and do not require any assumptions about the number or relative size of the underlying clusters.

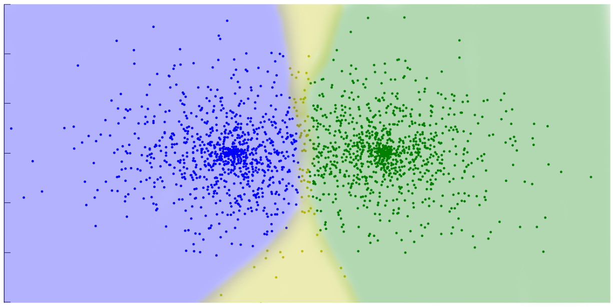

Fig. 5. Clustering of two Gaussians using α = 0.5, p = 0.5.

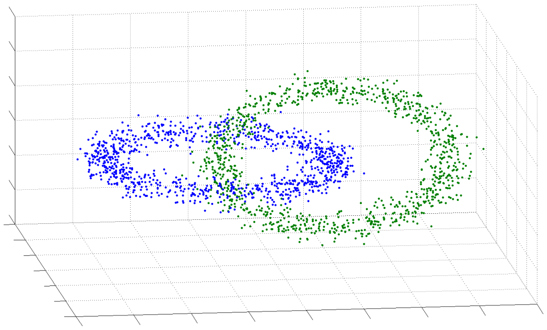

Fig. 6. Clustering of two rings using IsoClustering.

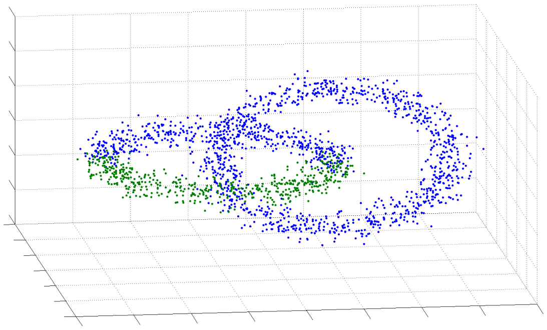

Fig. 7. Clustering of two rings using Spectral Clustering.

That these tests were done on weighted graphs was intentional, since overlapping communities on weighted graphs are not as frequently explored; but this is not meant to imply that the algorithm is limited to either weighted graphs or synthetic data. As an example of its performance on an unweighted graph constructed of real data, Table I and Fig. 9 show the results of the algorithm when used to partition the College Football network, which it does with high accuracy.

VI. Conclusions and Future Work

That the framework presented provides the means to establish precise definitions of clusters does not preclude its generality. We have shown results based on only a select few out of many possible interpretations for volume and perimeter, and even those select cases have produced competitive results. That particular choices result in its equivalency to well-known contemporary methods serves to further cement the inclusiveness of the overall framework. Furthermore, since the approach is wholly based on isoperimetric inequalities, it is also connected solidly with mathematical theory that can be the basis for future extensions.

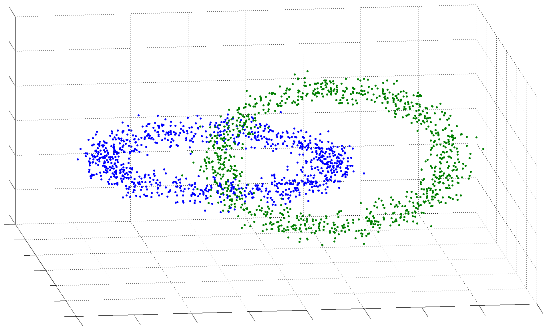

Fig. 8. Clustering of two rings using Cheeger Cut Clustering.

Football Teams	IsoClustering	AFA	Modularity
Atlantic Coast	1	1	0.9000
Big East	1	0.8000	1
Big10	1	1	1
Big12	1	1	0.9231
Conference USA	0.9000	0.6429	0.9000
IA Independents	0.4000	0	0
Mid-American	0.9231	0.8667	0.8667
Mountain West	1	1	0
Pac10	1	1	0.5556
SEC	0.9167	1	0.7500
Sunbelt	0.5714	0.4444	0.4444
Western Athletic	0.7000	0.7273	0.7273
Overall	**0.8957**	**0.7901**	**0.6723**

TABLE I. Purity Comparison of Detected Communities in US College Football Data Set, Using IsoCluster Partitioning, Attractive Force Algorithm [4], and Modularity [11].

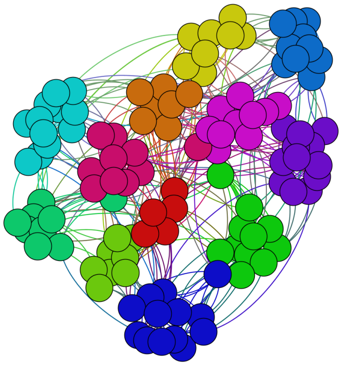

Fig. 9. Detected Communities with US College Football Data Set, Using IsoClustering Partitioning.

As the proposed cluster framework has been presented in its abstract form, it is then possible to extend its application to graphs governed by factors other than undirected similarity. For instance, once suitable definitions for V and P are chosen, the definition of clusters for directed graphs, multipartite graphs, and time-dependent graphs are made clear.

With the noted addition of the resolution parameter α, the enforcement of bottlenecks can be adjusted. In this way it is possible to both find clusters of varying size and number, as well as test for their stability. In future explorations, it would be of particular value if the relationship between α and sample deviation could be more precisely quantified.

Furthermore, the improved results on synthetic data produced by L^p volumes for values of $p < 1$ suggests that there is still room for exploration of volume definitions that result in better clustering than had been available previously. The exact manner in which this parameter affects the resulting clustering can be the subject of further research.

References

[1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[2] Z. Lu, W. Wu, W. Chen, J. Zhong, Y. Bi, and Z. Gao, "The maximum community partition problem in networks," *Discrete Mathematics, Algorithms and Applications*, vol. 5, no. 04, p. 1350031, 2013.

[3] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[4] Y. Hu, H. Chen, P. Zhang, M. Li, Z. Di, and Y. Fan, "Comparative definition of community and corresponding identifying algorithm," *Physical Review E*, vol. 78, no. 2, p. 026121, 2008.

[5] X.-S. Zhang, Z. Li, R.-S. Wang, and Y. Wang, "A combinatorial model and algorithm for globally searching community structure in complex networks," *Journal of combinatorial optimization*, vol. 23, no. 4, pp. 425–442, 2012.

[6] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[7] R. Osserman, "The isoperimetric inequality," *Bulletin of the American Mathematical Society*, vol. 84, no. 6, pp. 1182–1238, 1978.

[8] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[9] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[10] X. Bresson, T. Laurent, D. Uminsky, and J. V. Brecht, "Convergence and energy landscape for cheeger cut clustering," in *Advances in Neural Information Processing Systems*, 2012, pp. 1385–1393.

[11] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.