AVALIAÇÃO DA QUALIDADE DE AGRUPAMENTOS EM GRAFOS

HÉLIO MARCOS PAZ DE ALMEIDA

AVALIAÇÃO DA QUALIDADE DE AGRUPAMENTOS EM GRAFOS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Dorgival Olavo Guedes Neto

Belo Horizonte

Dezembro de 2012

HÉLIO MARCOS PAZ DE ALMEIDA

AVALIAÇÃO DA QUALIDADE DE AGRUPAMENTOS EM GRAFOS

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais - Departamento de Ciência da Computação in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Dorgival Olavo Guedes Neto

Belo Horizonte

December 2012

Almeida, Hélio Marcos Paz de
A447a Avaliação da qualidade de agrupamentos em grafos / Hélio Marcos Paz de Almeida. — Belo Horizonte, 2012
xxiii, 90 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação
Orientador: Dorgival Olavo Guedes Neto

1. Computação - Teses. 2. Teoria dos grafos - Processamento de dados - Teses. 3. Aglomeração - Teses. 4. Mineração de dados (Computação) - Teses. I. Título.

CDU 519.6*62(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Avaliação da qualidade de agrupamentos em grafos
(The evaluation of graph clustering quality)

HÉLIO MARCOS PAZ DE ALMEIDA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. DORGIVAL OLAVO GUEDES NETO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ALEXANDRE PLASTINO DE CARVALHO
Instituto de Computação - UFF

PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG

PROF. MOHAMMED JAVEED ZAKI
Rensselaer Polytechnic Institute

PROFA. SANDRA APARECIDA DE AMO
Faculdade de Computação - UFU

PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 14 de dezembro de 2012.

To Nathalia, with all my love.

Agradecimentos

Foi uma longa e árdua jornada que, felizmente, chegou ao fim. Sem dúvida uma vitória, mas seu mérito não é apenas meu. Muitas pessoas me ajudaram, direta ou indiretamente, a chegar aonde estou agora, e eu gostaria de registrar aqui meu agradecimento formal a essas pessoas.

Sem nenhuma ordem em particular, gostaria primeiro de agradecer a Deus e a minha família. Meus pais, Herbert e Fátima, e meu irmão, Herbert, pelo constante apoio e suporte. Também gostaria de agradecer ao resto da minha família, meus tios Leonardo, Vera, Ana, Calíope, Maria Teresa, Cristina, suas respectivas famílias e, em especial, à minha avó Militina. Por mais que eu esteja distante, sei que vocês sempre estiveram torcendo por mim, e vocês também sempre estiveram em meu coração. Também gostaria de agradecer aos meus tios, primos e primas de Belo Horizonte, em especial às minhas tias Alzira, Rosário e Irene, que me aceitaram de coração aberto e me fizeram sentir parte da família desde o momento que cheguei aqui. Sinto muito por ter ficado tão ausente, especialmente nos últimos anos do doutorado.

Uma pessoa muito especial e que, por sua vez, também merece um agradecimento especial é minha noiva, Nathalia. Ela sempre esteve ao meu lado, tanto nos momentos bons quanto nos (muitos) momentos (muito) ruins, me dando o apoio e encorajamento necessários para que eu conseguisse seguir em frente. Todos os dias, com seu jeito meigo e alegre, ela me faz lembrar o quanto eu a amo. Muito obrigado, Nath.

Também gostaria de agradecer a meus tios e tias de Belém, Matos, Gilka, Petrus, Dida, Alexandre e Amira. Meus padrinhos Reiko e Guilherme. Meus amigos Mário, Marilia, Nádia, Rosana, Alexander, Sueleny, Otávio e Pio. Alguns estiveram mais presentes durante todo o meu doutorado, mas sem dúvida todos estiveram ao meu lado, de uma forma ou de outra.

Gostaria também de agradecer formalmente aos professores Wagner Meira e Mohammed Zaki que, mesmo sem ter nenhuma ligação formal com a minha tese, me deram todo o apoio que eu precisei. Sem a ajuda deles eu com certeza não estaria onde estou, e por isso vocês tem minha eterna gratidão.

Outros que sempre estiveram ao meu lado foram os colegas e amigos do Speed. Em diversas avaliações sobre qual era a melhor coisa do laboratório, "as pessoas" sempre ganhava disparado. Eu não poderia concordar mais. Agradeço ao Fernando, Arlei e Pedro, que sempre me ajudaram nas minhas dúvidas mais técnicas, além de servirem de cobaias para as minhas ideias e apresentações. Ao Walter, Sílvio, Coutinho, Zilton e toda a garotada de IC pela companhia e bons momentos. Não poderia também deixar de agradecer aos colegas que já deixaram o speed: Tiago Macambira, Fireman, Paolo, Sachetto, George, Charles, André (Hawks), Orair, Rodrigo, Thatyene, entre muitos outros. Todos eles, de uma forma ou de outra, me encorajaram e deram apoio para que eu conseguisse terminar o curso.

Também gostaria de agradecer ao pessoal da Astrofísica, um laboratório que gosto de me considerar membro honorário: Francisco, Fábio, Bárbara, Alana, Mateus, Marcelo, Pauline, Gustavo e Wilson. E aos amigos Artur Jeber, Paulo e Denise. Sei que provavelmente esqueci de comentar o nome de alguns amigos nesses agradecimentos, então se você está lendo esse texto e se sentiu omitido, por favor assuma que foi só um lapso momentâneo meu.

Resumo

O processo de descoberta de grupos de vértices similares e conectados em um grafo, conhecido como agrupamento em grafos ou *graph clustering*, possui aplicações interessantes em diversos cenários, tais como biologia, marketing e sistemas de recomendação. Um dos grandes desafios da área de agrupamentos em grafos é a avaliação da qualidade dos agrupamentos, que é utilizada para medir a efetividade de algoritmos de agrupamento. Existem muitas métricas de qualidade para avaliação de agrupamentos em grafos, mas não há consenso sobre qual delas é melhor adequada para essa tarefa, e a maior parte dos autores na literatura simplesmente assume que uma métrica escolhida é boa o suficiente, com pouco ou nenhum interesse em avaliar a força dessas afirmações.

Para melhor compreender a efetividade das métricas de qualidade de agrupamentos mais populares apresentadas na literatura, estudamo-las em diferentes cenários. Descobrimos que essas métricas apresentam fortes tendenciosidades e inconsistências estruturais que fazem com que a qualidade de seus resultados seja, no mínimo, duvidosa. Nossos estudos demonstraram que, apesar dessas métricas de qualidade avaliarem corretamente a esparsidade de conexões entre grupos, elas não avaliam adequadamente a densidade interna dos mesmos, ignorando informações essenciais, como a número de vértices pertencentes a cada grupo, ou mesmo ignorando, na prática, métodos de avaliação de densidade interna devido ao seu alto custo computacional.

Tendo isso em mente, propusemos um novo método de avaliação da densidade interna de um dado grupo, um que não apenas utiliza informações mais completas na sua avaliação de densidade, mas que também leva em consideração as características estruturais do grafo de origem. Com esse método, a densidade interna de um grupo é avaliada em termos da densidade esperada de grupos similares oriundos do mesmo grafo. Isso difere das outras métricas disponíveis, onde grupos de diferentes grafos são comparados a partir dos mesmos parâmetros, um comportamento que penaliza redes que sejam naturalmente mais esparsas. Então, propusemos uma nova métrica de qualidade para agrupamentos em grafos, combinando nossa métrica de avaliação da qualidade interna e Condutância, uma popular métrica de avaliação de esparsidade

externa. Dessa forma, a métrica proposta avalia as duas principais características estruturais esperadas de grupos bem formados. Nossos experimentos mostraram que a a métrica proposta é capaz de penalizar corretamente grupos mal formados que seriam bem avaliados por outras métricas de qualidade presentes na literatura, ao mesmo tempo que concedem boas pontuações a grupos bem formados.

Abstract

The process of discovering groups of similar, connected vertices in a graph, known as graph clustering, has interesting applications in several scenarios, such as biology, marketing and recommendation systems. A major challenge concerning this problem is the evaluation of cluster quality, which is used to measure the effectiveness of clustering algorithms. Many quality metrics for graph cluster evaluation exist, but there is no consensus on which ones are best suited for this task, and most authors in the literature just assume that a chosen metric is good enough, with little or no interest in evaluating the strength of such claims.

To better understand the effectiveness of the most popular cluster quality metrics presented in the literature, we studied them in different scenarios. We discovered that they present strong biases and structural inconsistencies that cause the quality of their results to be, at least, doubtful. Our studies demonstrated that, while in general those popular quality metrics do a good job evaluating the external sparsity between clusters, they do poorly when evaluating their internal density, ignoring essential information, such as the cluster's vertex count, or having its internal density ignored in practice because of computational costs.

With that in mind, we proposed a new method for evaluating the internal density of a given cluster, one that not only uses more complete information to evaluate that density, but also takes into consideration structural characteristics of the original graph. With this proposed method, the internal density of a cluster is evaluated in terms of the expected density of similar clusters in that same graph. That is in contrast to the traditional quality metrics available, where clusters from different graphs are compared by the same standards, a behavior that penalizes naturally sparser graphs. Then, we proposed a new quality metric for graph clusters, combining our metric for internal quality evaluation and Conductance, a popularly used metric for external sparsity evaluation. This way, the proposed metric evaluates the two main structural characteristics expected from well formed clusters. Our experiments showed that the proposed metric is capable of correctly penalizing badly formed clusters that were

highly ranked by other quality metrics from the literature, while still awarding high scores for good ones.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

A graph is a mathematical model that allows a simple, yet powerful, representation of elements and their relationships. In a graph, the elements are represented by vertices, and their relationships as edges that link them. They can be used as a representation of any kind of network, even those based on real life entities, such as friendship, food webs, protein interactions, power grids or airline routes, to name a few. When studied as graphs, such real networks present some interesting characteristics that are not easily found in random or homogeneous networks, such as a great variation on the number of connections each element has, also known as their *degree*, and on the density of relationships in different parts of the network. Networks with such characteristics suggest that there is some sort of internal ordering, or an underlying structure that binds elements, and are named *complex networks* [Newman, 2003c].

When networks are mapped as graphs, it is possible to study them using existing graph theory tools and techniques. One specific area of graph studies, called *graph mining* [Chakrabarti and Faloutsos, 2006], uses those techniques to search graphs for unusual, interesting, or significant patterns in their structure. Such patterns can help us understand the overall behavior of a graph's elements, providing interesting insights about its structure and enabling, for example, the prediction of network growth or the discovery of influential elements in the network structure.

One of those interesting patterns that may exist in such networks is the occurrence of *clusters*, groupings of elements that are more similar among themselves than they are with the rest of the network. The sub-area of graph mining responsible for the discovery of such groupings is called *graph clustering*.

The automatic discovery of clusters can be interesting in many different scenarios. For example, recommendation systems can use the clustering of purchase relationships (clients ands product they buy) for better results [Reddy et al., 2002]; clustering Web

clients by their interests and their network distances can be used to optimize server usage [Krishnamurthy and Wang, 2000]; Web pages can be clustered to help identify common topics and structures formed by several interconnected documents [Wong and Fu, 2000]; in biology, clustering can be used to help in the classification of gene expression data [Xu et al., 2002] and protein interactions [Pereira-Leal et al., 2004; King et al., 2004], among many other possibilities.

However, discovering meaningful clusters in a graph is not a simple task. There is no universally accepted definition of what should be the structure of a well-formed cluster, but the most adopted and classical view is based on the concept of homophily, which states that similar elements have a greater tendency to group with each other than with other, less similar elements [Newman, 2003b]. When working with the structure of edges and vertices only, similarity between vertices can be given by their minimum distance, with closer vertices being more similar, for example. However, the most accepted representation of similarity for simple, undirected and unweighted graphs is evaluated in terms of edge densities, with a good cluster having more edges linking its own elements among themselves (i.e., it has high internal edge density) than linking them to the rest of the graph (i.e., it has sparser external connections).

The concept of edge densities being good descriptors for vertex similarity is very well accepted in the literature, but discovering such edge-dense clusters in graphs is a complex task. By this definition, cluster structure can be anything between a connected subgraph and a maximal clique (an NP-complete problem), with the better results leaning towards the latter [Schaeffer, 2007].

Since the problem of clustering does not have an exact solution, many heuristics have been proposed to find clusters which maximize both intracluster density and intercluster sparsity. Examples of such algorithms are MCL [Dongen, 2000, 2008], K-means [Steinbach et al., 2000a] and Spectral [Kannan et al., 2004; Schaeffer, 2007]. However, those heuristics are not guaranteed to present optimal, or even good, clustering results.

When a graph is small, with only a few vertices and edges, clustering results found for it can be evaluated manually. For this kind of evaluation, an expert in the subject represented by the graph evaluates each cluster by hand, identifying if they make sense. This kind of evaluation has highly precise results, as expert knowledge and human reasoning are very good to detect any subtleties that the structures found might have and that have influence on cluster quality. However, as the size of the graph grows, manual evaluation becomes unfeasible. In those cases, quality metrics that score a given cluster, or even a full clustering, considering characteristics expected to exist in well-formed clusters, may be used as quality indicators. The scores obtained through

those metrics can be used to infer the quality (or optimality) of a given clustering, or as a basis of comparison for different clustering results. Some clustering approaches even use heuristics to look for clusters that might maximize those metrics, since finding clusters by optimizing those quality metrics is computationally complex [Šíma and Schaeffer, 2006; Brandes et al., 2008a; Shamir et al., 2004]. Examples of such metrics are modularity [Newman and Girvan, 2004] and conductance [Kannan et al., 2004], which evaluate clusters using only their structural characteristics.

But do those quality metrics really provide an accurate view of what a good clustering should look like? There is no easy answer for this question. In order to evaluate the quality of results obtained by those metrics, it would be necessary to know what is the best clustering of a given graph and observe how the studied metrics score this case. However, those metrics themselves were created because identifying a graph's best clustering is a hard problem, making the evaluation of quality metrics a problem as hard as evaluating the clusters themselves.

So, most of the evaluation done on the quality of those metrics was done on graphs where the expected clustering was known beforehand. The only problem with this solution is that most graphs that have a known solution are relatively small and present very well-formed clusters. To assume that positive results obtained for those kinds of networks can be easily extrapolated for larger, more complex networks without any loss of generality is, at best, naive. However, that is what most of the works in the literature do. More than that, most of the subsequent clustering works simply assume that the existing metrics are good enough and use them indiscriminately.

We, however, consider the evaluation of cluster quality to be one of the most important problems in the area, since it gives, when done correctly, the most approximate view of what the ground truth looks like for a network with unknown expected clustering. So, answering the question of the effectiveness of currently used quality metrics for graph clustering is essential.

To answer this question, we studied and compared some of the most popular clustering quality evaluation metrics used in the literature. We have done so by observing the behavior of those quality metrics when applied to groupings of real world networks obtained through classic clustering algorithms, with different levels of granularity. Our goal was to identify if those metrics correctly represented the classical structure of a good cluster (internally dense and externally sparse), especially when applied to larger, less predictable networks. What we discovered is that those popular quality indexes have strong structural anomalies in their formulations, which cause them to be biased and unreliable, a behavior that is more pronounced when those metrics are applied to results from larger, real world graphs.

With that knowledge in mind, we decided that there was a need to find other, more accurate ways to evaluate cluster quality. So, we extend our evaluation of cluster quality metrics in order to identify what causes the metrics considered to present any biased behavior. We concluded that those metrics do not correctly evaluate one of the two key elements of cluster quality: its internal density. Another problem identified was that those metrics evaluate all clusters by the same standards, causing clusters from naturally sparser graphs (such as technological networks, for example) to be unfairly penalized.

Hence, we proposed a new method to evaluate a cluster's internal density. Our method uses more complete information to evaluate a cluster's internal density. Additionally, it uses the characteristics of the studied graph in order to discover what are the density thresholds that identify interesting clusters for that particular case. This allows for a more "local" evaluation, removing the penalization that sparser graphs usually receive from current quality metrics. Using this new internal quality evaluation metric, we then proposed a new quality metric, mixing both our new metric and conductance, an external sparsity evaluation index, in order to build a new, more effective quality metric for graph cluster evaluation. We show thorough our experiments that our proposed metric correctly evaluates the quality of clusters which would be incorrectly evaluated by usual metrics, such as modularity.

1.1 Statement of Thesis

Currently used quality metrics for evaluating graph clusters cannot correctly evaluate the two main characteristics of a good cluster, namely internal cluster density and inter-cluster sparsity, at the same time. To correct that, we have studied some of the most popular clustering quality evaluation metrics used in the literature, in order to identify how to solve their limitations.

Based on the results obtained from this study, we have proposed a novel quality metric that not only better evaluates both cluster characteristics, but also is effective when applied to clusterings of different types of networks.

1.2 Contributions

The main contributions of this work are:

- An in-depth study of currently used quality metrics for graph clustering.

- Analysis on the shortcomings of those metrics.
- Proposal of a new technique to evaluate internal cluster density, one of the two main characteristics of a well-formed cluster.
- A new quality metric to evaluate cluster quality in graphs.

1.3 Organization of the Text

The rest of this work is organized as follows: Chapter 2 describes in more detail the concept of graph clustering, also presenting some of the most important clustering algorithms. The most well known cluster quality evaluation metrics are described in Chapter 3. In Chapter 4, we detail the scoring behaviors and biases of those quality metrics through theory and experimentation. Based on the knowledge obtained from our experiments on the previous chapter, in Chapter 5 we propose a new quality evaluation metric, which we test through experiments shown on Chapter 6. Conclusions and possible future works are discussed in Chapter 7

Chapter 2

Graph Clustering

In this Chapter, we present the basic concepts of graph clustering. First, we discuss what exactly is a cluster and the connection of this concept to the so-called complex networks. Later, we discuss some of the techniques used to solve the problem of automatically discovering cluster structures in graphs.

2.1 Graph Structure and the Existence of Communities

A graph is a mathematical structure that can be used to model and represent the elements of a collection and their pairwise relationships. It can be described as a pair $G = (V, E)$, where V is the set of elements that belong to the collection, which are called *vertices*, and E is the set of pairwise relationships between the elements in V ($E \subseteq V \times V$), which are called *edges*. So, an edge (u, v) connects two vertices, u and v. The number of edges that touch a given vertex is said to be its *degree*.

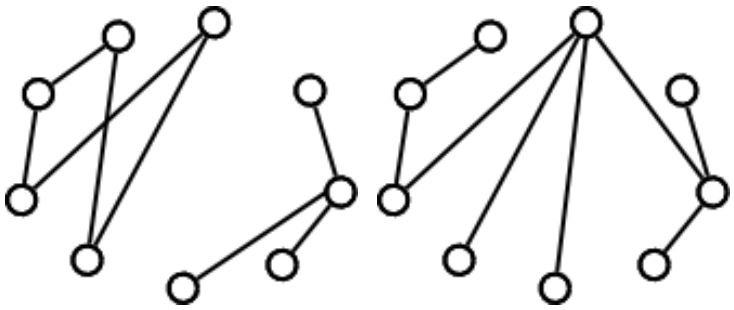
Vertices and edges might have varying levels of additional information that is linked to them, and this information defines the type of the graph. In a *simple* graph, only one edge may link two given vertices, and no edge may have the same vertex as source and target (i.e., $\nexists \{u, u\} \in E$). In a *weighted* graph, each edge has a weight value, which represents the strength of that connection. For *undirected* graphs, there exists an edge symmetry so that $\forall\ u, v \in V$, if $(u, v) \in E$ then $(v, u) \in E$. If this symmetry does not exist, than the graphs is said to be *directed*. Also, vertices and edges may have labels that describe their intrinsic characteristics, categories or functions. If that happens, than the graph is said to be *labeled*.

The relationships between vertices might cause the occurrence of interesting struc-

tures, such as the formation of communities or clusters. But is the occurrence of such structures random or does it reflect some kind of fundamental ordering existent in that network? To answer this question, we will briefly discuss how the structure and formation of a given network might influence the existence of communities in it. Later, we will discuss the structure of those clusters, specially on the kinds of graphs studied in this work.

2.1.1 Random Networks

The concept of random network or graph was first proposed by Solomonoff and Rapoport [1951] and, independently, by Erdös and Rényi [1959]. According to it, there is a p probability for an edge to link any two of the V vertices in a random graph. For low values of p, the graph will posses only a few edges, and nodes will form a large number of small, disjoint components. On the other hand, for high values of p, many of those small components merge, generating larger components. The component with the highest number of vertices is called the *giant component* or the *largest connected component* (LCC).



(a) Random network (b) Complex network

Figure 2.1: Examples of random and complex networks. Both have small distances between vertices, but complex networks have skewed degree distributions – few vertices have high degrees while most of them have low degrees.

This kind of graph has some interesting characteristics. The average shortest edge distance between two vertices in the same component is relatively smaller than what would be expected, considering the size of the whole component (Figure 2.1a). More than that, random graphs are very resilient to edge removals, which means that it is hard to break its giant component into two smaller ones by randomly, or even strategically, removing edges or vertices from it.

Although this graph generating model is simple, it still represents correctly at least one characteristic of real networks: the small hop distance between vertices. This

characteristic can be seen, among other places, in the famous "small world" experiment by Travers and Milgram [1969], where a group of people received letters addressed to a man living in another region of the USA, and they were to try to deliver that letter only through people they knew.

2.1.2 Complex Networks

Although one important characteristic of real networks is accurately represented by random graphs, that model is far from perfect. More thorough studies of real networks showed that they possess other characteristics that could not be mapped by this kind of graph. Among those were power law vertex degree distributions and high clustering coefficients [Newman, 2003c], which will be described shortly. Graphs sharing those characteristics can be collectively called *complex networks* [Newman, 2003c], and examples can be found in diverse types of networks, such as social (e. g., e-mail networks [Ebel et al., 2002], instant messaging networks [Smith, 2002]), information (e. g., scientific paper citation networks [Egghe and Rousseau, 1990], World Wide Web [Huberman, 2001]), technological (power lines [Watts and Strogatz, 1998a], airline routes [Amaral et al., 2000], etc) and metabolic (e. g.,metabolic networks [Jeong et al., 2000], protein interactions [Ito et al., 2001]).

The power law degree distribution of a complex network means it is heavily skewed. This is very different from what happens in random networks (Figure 2.1b). In complex networks, few vertices possess high degrees, while most of them present low degrees, a behavior also known as "heavy tail". This characteristic also causes small hop distances between vertices, but in a different way than with the random model: in random networks, the random connections tend to create well distributed "pathways" in the graph, allowing easy connections between all nodes, while in complex networks, the high degree vertices, also known as *hubs* or *authorities*, are connected among themselves and to almost all other nodes and serve as shortcuts. Those hubs, however, are the backbone of the graph, and their removal easily disconnects the whole graph, making complex networks not as resilient as random ones.

The clustering coefficient evaluates how connected groups of vertices in a graph are. It does so by measuring the proportion of "triangles", trios of directly connected vertices (3-cliques), that exist in the network. Evidences shown in the literature suggest that complex networks tend to possess high values of clustering coefficient, specially when compared to Erdös-Rényi random graphs with of similar size and build [Newman, 2003c]. So, it is only fair to acknowledge that complex networks tend to have a higher level of cohesion, at least in some of their parts, and that this cohesion might form

community structures with a higher probability.

Also, it is valid to point out that new models for random graph generation were proposed, trying to emulate those newfound characteristics from complex networks. The Barabási-Albert model [Barabási and Albert, 1999], for example, adds power law degree distribution to random graphs by using the mechanism of *preferential attachment*, where the probability to create new links to vertices of high degree is higher. The Watts-Strogatz model [Watts and Strogatz, 1998b] adds high clustering coefficient by using a regular lattice as the initial structure of the graph, and then randomly rewiring part of its edges. Another model, by Newman [2009], allows the generation of graphs with guaranteed cluster structures. Nevertheless, even though those new random graph generation models are able to represent some characteristics from real networks, they are not guaranteed to represent all of them. For this reason, we favor the use of real graphs in all experiments done in this work.

2.1.3 Structure of a Cluster

The relationships between vertices might cause the occurrence of interesting structures, such as the formation of communities or clusters. As explained in Chapter 1, a cluster is a subset of vertices that present a remarkably high level of similarity among themselves and dissimilarity to the rest of the graph. But, as was shown earlier in this Chapter, a graph's elements might present different characteristics that add extra information that enriches it, and those extra layers of information have strong influence on the expected structure of clusters.

When working with simple and undirected graphs, for example, vertex similarity is usually evaluated in terms of edge densities. This way, a well-formed cluster is expected to be formed by vertices who are densely connected among themselves, while at the same time sparsely connected to the rest of the graph [Schaeffer, 2007]. However, this concept is so broad that a cluster's structure can be anything between a connected subgraph and a maximal clique, with the better results leaning towards the latter.

For the case of graphs with richer structures, the problem of defining the structure of a cluster grows. For example, if the graph is weighted, how do you consider the proper importance of an edge's weight? Can a cluster be considered externally sparse if it has only one edge connecting it to the rest of the graph, but this edge is ten times stronger than any other edge in the graph? If the graph is directed, can it be considered internally dense if it is highly connected, but not all vertices are reachable by all the others?

So, defining adequate ways of using the extra information on dimensions of a

graph in order to define the concept of similarity between vertices is very complex. Considering this, and also the fact that the quality evaluation metrics currently used in the literature for simpler graphs already struggle to do so effectively [1], in this work we will focus our efforts on the edge structure of simple, undirected and unweighted graphs as the only means to identify the level of similarity between vertices.

2.2 Clustering Techniques

Since the problem of clustering does not have an exact solution, many heuristics have been proposed to find clusters which maximize both intracluster density and intercluster sparsity. In this section, we will present some of the most important techniques used to solve the problem of automatic discovery of clusters in graphs. First, we will discuss the similarities and differences between clustering and partitioning, which are related problems. Then, we will describe some implementations of graph clustering, providing more detail for those that were used in our experiments. For a better presentation of the algorithms, we will divide the clustering techniques in two main groups: topological and semantic.

2.2.1 Clustering vs. Partitioning

In the literature, the problem of graph clustering is identified in two ways: partitioning and clustering. In partitioning problems, the number of clusters to be found must be known beforehand, and the clusters found should have roughly the same size. One example of partitioning algorithm is METIS [Karypis and Kumar, 1998], which uses a multilevel approach for partitioning, "coarsening" the graph by removing its edges until the graph is small enough to be easily partitioned and then "uncoarsening" it back to its original form, while keeping the cohesiveness of the clusters found.

For clustering problems, the number of clusters may or may not be a parameter, and there is no restriction on cluster size relationships. Some examples of clustering algorithms will be discussed in the following sections.

Partitioning is useful, for example, in situations where it is necessary to divide a graph for parallel processing, keeping in mind the problem of load balancing and reference locality of the data. One such work is that of Sarkar and Moore [2010], who uses the partitioning of the graph to help minimize the number of memory page faults during their algorithm's execution. However, because the partitioning algorithms focus

[1] This will be discussed on Chapters 3 and 4

on finding clusters with similar sizes, and not the intrinsic communities existent in the graph, they escape the focus of this thesis and will not be further discussed.

2.2.2 Topological Clustering

This category of clustering algorithms uses only the information obtained through the relationships between the vertices of the graph. They use typical graph theoretic techniques such as cuts, maximum flow and shortest paths to derive vertex similarity.

Topological clustering techniques can be grouped in several ways, such as by their clustering approach (top-down or bottom-up), the kind of graph they can handle (with directed and/or weighted edges, for example), or the locality of the information used for clustering (local or global). Here we will group them in some classic "families" of algorithms, as it is our belief that this will make for a simpler, more concise presentation.

2.2.2.1 Spectral clustering

Spectral clustering is a technique that uses the eigenvectors (spectrum) and eigenvalues of a graph's adjacency matrix to define cluster membership [Kannan et al., 2004; Schaeffer, 2007; Nascimento and de Carvalho, 2011] . It is based on the fact that if a graph is formed by k disjoint cliques, then it's normalized Laplacian will be a block-diagonal matrix with eigenvalue of zero, having multiplicity k, with the eigenvectors functioning as indicators of cluster membership. More than that, small perturbations like adding a few edges linking clusters or removing edges from inside the clusters will make the eigenvalues become slightly more than zero and change the eigenvectors, but not enough to cause the underlying structure to be lost. This clustering technique requires the number of desired clusters as an input.

The basic spectral clustering algorithm assumes the graph to be undirected, but some work has been done to extend it to allow the clustering of more complex graphs. Meilă [2007] adapts the spectral clustering paradigm so that it can deal with directed and weighted graphs. She uses teleporting random walkers to obtain the transition matrix needed by this kind of algorithm without having problems with of vertices without any outgoing edges (also called sinkhole vertices).

Another extension of spectral clustering was proposed by Zhou et al. [2005] to deal with directed graphs. This extension can also be used to classify unlabeled vertices in a partially labeled graph.

2.2.2.2 Cut-Based Methods

Another way to divide a graph into clusters is by recursively finding the minimum cut of a graph and to bisect it by removing the edges from this cut until doing so does not result in a better quality clustering or a given set number of clusters is found. There are many ways to find the best possible cuts. One is to use the *betweeness* of the edges of the graph. The betweeness is a measure of the amount of minimum paths between all vertices of the graph that use a given vertex or edge. So, an edge with high betweeness is one used by many of the possible minimum paths and, therefore, it may probably be connecting two different clusters. Iteratively removing the edges with highest betweeness until the graph is disjoint is the basis for the Betweeness Cut algorithm [Girvan and Newman, 2002].

Another method using the same idea of clustering by cuts is the Iterative Conductance Cut algorithm [Kannan et al., 2004]. It tries to find minimum conductance cuts to divide the clusters from each other, while keeping the clusters themselves with high values of conductance, since it means that those clusters are too internally dense to have cheap cuts and, therefore, are structurally good. The formal definition of conductance is described in more detail in Chapter 3.

2.2.2.3 K-means

In the traditional K-means algorithm [Hartigan and Wong, 1979], k elements are randomly chosen as the centroids of each one of k clusters to be found and other elements closer to a given centroid than to others are added to that cluster. With this basic cluster at hand, a new centroid is calculated for each cluster, reflecting their new "centers", and the process is repeated until the centroids calculated do not change anymore. It is important to note that K-means is not bound to a graph representation of data, since the "closeness" of two data elements can be derived in any way that is consistent with their representation, like euclidean distance for points in a 2-D plane or cosine similarity for text documents. When applied to graphs, the closeness between two vertices is given by the edge distance between them.

Bisecting K-means [Steinbach et al., 2000a] differs from the traditional algorithm in the following way: the whole graph is considered to be a cluster, which we bisect using traditional K-means, using the topological (edge) distance between the nodes. One of the new clusters is chosen to be once more bisected and the process repeats until the desired number of clusters is found. In general, the biggest remaining cluster is the one chosen for further partitioning, although other metrics can be used to guarantee that an already structurally sound cluster will not be arbitrarily broken down.

Zhou et al. [2009] present a K-means variant for labeled vertex graph clustering. In that paper, possible label values are transformed into virtual vertices that will have edges linking them to all vertices that possess the given label. A similarity matrix, which presents the similarity values between all vertices of the graph, for this extended graph is created using random walks to evaluate how close (*i.e.*, similar) the vertices are to each other. This matrix will be used as the basis for the traditional K-means algorithm.

2.2.2.4 Markov Clustering

The Markov clustering algorithm (MCL) is based on the simulation of stochastic flows in a graph [Dongen, 2000, 2008]. The basic idea behind MCL is that the distances between vertices are what identify a cluster, with small distances between vertices indicating that they should belong to the same cluster and large distances meaning the opposite. By that logic, a random walker would have greater probability to stay inside a cluster than to wander to neighboring ones, and the algorithm explores that to identify clusters.

The clustering process of MCL consists in two iterative steps: expansion and inflation. The expansion corresponds to the random walk itself, done by calculating the power of the normalized adjacency matrix that represents the graph, using traditional matrix multiplication. The inflation step consists in taking the Hadamand power of the expanded matrix, followed by a scaling step to make the matrix stochastic again, with the elements of each column corresponding to a probability value. MCL does not need to have a pre-defined number of clusters as input, it's only parameter being the inflation value, which affects the granularity of the clustering (the higher the value, the finer is the granularity).

Satuluri and Parthasarathy [Satuluri and Parthasarathy, 2009] extends the MCL algorithm for better scalability. They use a coarsening or simplification phase, similar to the one used in METIS and discussed previously, to obtain a smaller and simpler version of a large graph and then cluster it using stochastic flows. After that, the graph is "uncoarsened" back to its original form, but keeping the discovered cluster structure coherent in the process.

2.2.2.5 Modularity-Based Methods

Modularity is a quality metric for clusterings that gives high values for clusters whose number of internal edges surpasses the number of expected edges for a random graph of the same size. The details on how to calculate this metric are detailed in Chapter 3.

However, clustering searching for the optimal value of modularity is known to be a NP-complete problem [Brandes et al., 2008a]. So, the Fast Modularity algorithm [Newman, 2003a] is a greedy optimization solution for cluster modularity. In the beginning, each vertex is part of a different cluster. At each step, the two clusters whose combination would result in the greater modularity gain (or the lowest modularity reduction) are merged in a single cluster. In the end, the result will be the configuration that gives the best modularity score during the process.

The original modularity formulation is only applicable to undirected graphs. Leicht and Newman [2008] propose a modularity-based algorithm for directed graph clustering. In this case, modularity will be adapted to consider the direction of edges in the following way: consider two vertices A and B, where A has high out-degree and low in-degree, and B has the reverse configuration. If there is an edge going from B to A, since this configuration is uncommon given the degree distribution of those two vertices, it should weight more in the modularity calculation in the same way that the number of "expected" edges inside a random graph were used in the classic modularity. Then, it will use the eigenvalues and eigenvector of the modularity matrix, which is a matrix with the modularity values each pair of vertices in the graph contribute to the overall modularity of the clustering, to discover the best possible configuration, modularity-wise. Since this calculation requires a symmetric matrix, the modularity matrix will be summed with its own transpose.

2.2.2.6 Other Approaches

There are many other approaches that do not fall into any of the more traditional clustering methods. One of those is described by Lu et al. [2009], which uses the simulation of a "naming game" to define the clustering. In this game, every vertex starts with a single, random word in its vocabulary. With each step of the game, one vertex tries to communicate one of the words it knows to one random neighbor: if the neighbor didn't have the word in its vocabulary, it will add it; if it already had the word, both vertices would discard all other words from their vocabularies, except for the common one. The authors show that, if there is a community structure in the graph, this game won't converge, but if it is let running for a couple of rounds, the densest communities will agree on one single word each.

The SCAN algorithm [Xu et al., 2007] uses a pairwise similarity metric based on the number of common neighbors between two vertices to group them. If this similarity is greater than a given threshold, those two vertices merge and start a new cluster, trying to add their neighbors until no neighbor is similar enough to be added to

that cluster. This algorithm also has the notion of vertices that do not have similarity enough to be part of any cluster, which are called hubs if they are connected to more than one cluster, and outliers if they are linked to only one cluster.

Another technique is presented by Bagrow and Bollt [2005]. Here a vertex is randomly chosen to be the seed of a cluster and it starts to grow this cluster adding its neighbors to it. This process grows the cluster, adding the next neighbors to the cluster like ripples in a pond, while the result of the division between the edges linking the border vertices with the rest of graph and this same value on the previous round is lower than a *alpha* value, which is an algorithm parameter. This approach clearly has problems with the choice of the seed, since the extremely regular evolution of the cluster might cause problems with more irregularly shaped clusters, so multiple executions are necessary for a more trustworthy answer.

The work by Palla et al. [2005] uses the idea of k-cliques as the basis for the clustering. To do this, they have to first discover all cliques existent in the graph, in decreasing order of size. Then they try to merge those cliques using the notion of "k-clique template rolling", where you first get one base clique and then try to "roll" it to a neighboring cluster, keeping one vertex fixed and trying to "move" the rest of the clique template to its other neighbors. This action that can only be done if the second clique has $k - 1$ vertices in common with the first clique and k is the size of the first clique. One interesting detail about this technique is that it allows for *cluster overlapping*, meaning that one vertex can be a member of more than one cluster at the same time.

2.2.3 Semantic Clustering

Sometimes the graph to be clustered has more information available than just its topological features, such as vertex labels that describe intrinsic characteristics of it. In cases like that, even ignoring the underlying structure linking its elements, a graph's vertices can still possess enough similarity between themselves to be logically grouped. Their similarity will be derived from other characteristics, such as cosine similarity for their contents in the case of text documents or categorical attributes for example. Here we will discuss some different algorithms used to cluster this kind of data.

The K-means algorithm [Hartigan and Wong, 1979] can be adapted to work with real world data containing categorical values. One of those adaptations [Huang, Zhexue, 1998] uses a simple matching dissimilarity measure to deal with categorical objects, replacing the means of clusters with its modes, and using a frequency-based method to update modes in the clustering process to minimize the clustering cost

function. This way, it can group elements who share common attributes.

Another algorithm used for semantic clustering is CURE [Guha, S., 2001]. It achieves the clustering of a dataset by merging elements into clusters until the desired number of clusters is found. It starts with every element being part of a single, different cluster (a singleton). Then, a certain fixed number of well scattered (distance-wise) elements are chosen. Those elements will be "shrunk" toward the center of their nearest cluster, effectively merging them to it. The authors of this work argue that this process helps to avoid the negative effects caused by outlier elements in the dataset.

Yet another algorithm used in this kind of clustering is BIRCH [Zhang, Tian et al., 1996]. It is a clustering method for very large datasets, making a large clustering problem tractable by concentrating on densely occupied portions of the dataset and using a compact summary of that information for quick evaluation. This summarization also removes outlier data and is stored and is incrementally updated in a height balanced tree. BIRCH then utilizes a traditional clustering algorithm, such as K-means, on the summarized tree to obtain the desired clustering.

Other proposed algorithm is ROCK [Sudipto Guha and Shim, 1999]. In it, two elements are said to be *neighbors* if they have a similarity value, that can be given by the Jaccard Index for categorical data, greater than a given threshold. Elements that share many neighbors are said to have strong *links* and their merging into a single cluster will result in better, more meaningful clusters.

2.2.4 Hybrid Clustering

Not many works approach the problem of clustering data using both a graph's topological structure and the semantic derived from its labels. One of them is presented by Lappas et al. [2009], where they try to discover optimal work groups for a given task. It uses a graph where the vertices are people, the edges link two people that have a good work relationship, and a vertex's labels indicate a person's proficiencies. The paper proposes some heuristics to discover the best possible groups to solve the problem, which means a set of vertices that have all needed proficiencies and are reasonably well connected to each other.

Another work is the one by Zhou et al. [2009], where the semantic information is incorporated into the graph topology by transforming all labels into virtual vertices that are linked to all vertices that possess that label. With this extended graph in hand, it uses traditional topological cluster to obtain the groupings. They also use entropy as a way to measure the coherence of labels inside a cluster.

Chapter 3

Quality Metrics

There is no consensus on what the main characteristics that define a good cluster are, but the most accepted view is based on the concept of assortative mixing, which states that elements have a greater tendency to form bonds with other elements with whom they share common traits than with others [Newman, 2003b]. In other words, the structure of a good cluster depends on two main characteristics: cluster elements should be highly similar among themselves, while at the same time being highly dissimilar to the other elements of the set. This similarity can be defined in many forms. For example, if the studied set is formed by data points placed on a Cartesian plane, similarity can be given by the Cartesian distance between the data points. This kind of similarity is used by algorithms such as the KNN [Laboratories et al., 1966].

When clustering is applied to graphs, the idea of high internal similarity and external dissimilarity persists, but the concept of similarity must change in order to accommodate the inherent structure of this kind of data. In this new environment, element similarity may be derived from different edge or vertex characteristics, such as edge density [Girvan and Newman, 2002], vertex distance [Tan et al., 2005] or labels [Zhou et al., 2009].

In this chapter, we will present some of the most popular graph cluster quality metrics in the literature. Some of them use only a cluster's structure to measure quality, while some use satellite data, such as labels, and others use a combination of both. Since the main focus of this work is the study and evaluation of cluster quality metrics for simple, undirected and unweighted graphs, a greater focus will be given to quality metrics that use only structural information.

3.1 Graph Definitions

A *graph* $G = (V, E)$ is composed of a set V of *vertices* and a set $E \subseteq \{V \times V\}$ of *edges*. If nothing is said against it, we assume that the graphs discussed are undirected, so E is symmetric. The number of edges of a graph G is $|E(G)| = m$, and the number of edges linked to a given vertex v (its *degree*) is represented as $deg(v)$. Edges may have an associated weight $w(u, v)$. In unweighted cases, we assume that $w(u, v) = 1 \forall (u, v) \in E$.

A *clustering* C is the set of all *clusters* of a graph, so that $C = \{C_1, C_2, \ldots, C_k\}$, and the number k of clusters may be a parameter of some clustering algorithms. Also, unless stated otherwise, $C_i \cap C_j = \varnothing, \forall i \neq j$. A cluster C_i that is composed by only one vertex is called a *singleton*, and $\bar{C}_i$ is the set of all vertices not in C_i. The weight of all internal edges of a single cluster is given by $w(C_i)$, a shortcut for $\sum_{e \in E(C_i)} w(e)$, where $E(C_i) = \{(u, v) \in E | u, v \in C_i\}$. By the same logic, $\bar{w}(C)$ is the sum of the weights of all inter-cluster edges.

Consider $E(C_i, C_j) | i \neq j$ as the set of edges linking clusters C_i and C_j. Also, $E(C)$ is the set of all internal edges for all clusters in C, and $\bar{E}(C)$ is the set of all *inter-cluster edges* in the graph $\{(u, v) | u \in C_i, v \in C_j, i \neq j\}$.

A graph cut $K = (S, \bar{S})$, where $\bar{S} = V \setminus S)$, divides a set of vertices V into two disjoint subsets ($S \cap \bar{S} = \varnothing$). The *cost* of a cut is given by the sum of the weights of the inter-cluster edges. Another important concept is that of an *induced graph*, which is a graph formed by a subset of the vertices and edges of a graph so that $G[C_i] = (C_i, E(C_i))$.

3.2 Topology-Based Metrics

One possible way to evaluate similarity between vertices of a graph is to observe the strength of their relationship considering the edges that connect them. Quality metrics that use this kind of information are called *topological*, and in this section we will present some of the most popular metrics of this kind used in the literature.

3.2.1 Modularity

One of the most popular validation metrics for topological cluster evaluation, modularity states that a good cluster should have a bigger than expected number of internal edges and a smaller than expected number of inter-cluster edges when compared to a random graph with similar characteristics [Newman and Girvan, 2004]. The modularity score Q for a clustering C is given by Equation 3.1, where e is a symmetric

matrix whose element e_{ij} is the fraction of all edges in the network that link vertices in communities i and j, and $Tr(e)$ is the trace of matrix e, i.e., the sum of elements from its main diagonal, and the operator $||e||$ represents the sum of all elements from a matrix e ($\sum_{i=1}^{n}\sum_{j=1}^{n} e[i,j]$).

$$Q(C) = Tr(e) - ||e^2|| \tag{3.1}$$

The modularity score Q often has values between 0 and 1, with 1 representing a clustering with very strong community characteristics. Some specific cases can have negative values, such as when many singleton clusters exist. Since those singletons contribute with external density, but not with internal density, they cause a strong imbalance on the modularity calculation that may lead to negative values.

$$Q(C) = \sum_{c \in C} \left[\frac{|E(c)|}{m} - \left(\frac{\sum_{v \in c} deg(v)}{2m} \right)^2 \right] \tag{3.2}$$

The formula presented in Equation 3.1 is the most classical one. However, it has a high computational cost, requiring a matrix multiplication. To avoid that cost, some works propose different, simpler ways to calculate modularity. One of those variants, proposed by Good et al. [2010], can be seen in Equation 3.2. This new formulation is cheaper to compute, but its results are only similar, not equal, to the ones obtained with the original formula. Because of this difference of results, unless otherwise noted, any reference to modularity in this text refers to the original formulation (Equation 3.1).

It is important to notice that those formulations assume the graph to be simple, undirected and unweighted. One example of adaptations for other classes of graphs can be found in the work by Leicht and Newman [2008], where the authors create a variation of modularity to be used with directed graphs. However, since the main focus of our study is on the evaluation of clusters from simple, undirected graphs, further discussion on those expansions falls out of our scope.

3.2.2 Silhouette

Silhouette uses the distance between vertices as a measure of their similarity [Tan et al., 2005]. It uses concepts of cohesion and separation of clusters in order to evaluate them. The silhouette index assumes that two vertices are more similar if they have a low minimum hop-distance between them, and that a good cluster should be formed by vertices which have a low average distance between themselves (internally cohesive) and a high average distance to vertices outside the cluster (externally separate).

The silhouette index is computed for each vertex and averaged for each cluster and/or the full clustering. Equation 3.3 is used to obtain a node i's silhouette index, where a_i is node i's average distance to all nodes inside its own cluster and b_i is the lowest average distance from node i to a cluster that is not it's own.

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{3.3}$$

Silhouette values can vary between -1 and 1. Negative value are undesirable, as they mean that the average internal distance is greater than the external one, making it less cohesive.

3.2.3 Conductance

The conductance [Kannan et al., 2004] of a cut is a metric that compares the size of a cut (i.e., the number of edges cut or the sum of their weights) of the edges in either one of the two sub-graphs induced by that cut. The conductance $\phi(G)$ of a graph is the minimum conductance between all of its clusters.

Consider a cut that divides G into k non-overlapping clusters $C_1, C_2 \dots C_k$. The conductance of any given cluster $\phi(C_i)$ can be obtained as shown in Equation 3.4, where $a(C_i) = \sum_{u \in C_i} \sum_{v \in V} w(u, v)$ is the sum of the weights of all edges with at least one endpoint in C_i. This $\phi(C_i)$ value represents the cost of one cut that bisects G into two vertex sets C_i and $V \setminus C_i$. Since we want to find a number k of clusters, we will need $k - 1$ cuts to achieve that number. In this paper we assume the conductance for the whole clustering to be the average value of those $(k - 1)$ ϕ cuts, as formalized in Equation 3.5. Conductance values computed this way vary between 0 and 1.

$$\phi(C_i) = \frac{\sum_{u \in C_i} \sum_{v \notin C_i} w(\{u, v\})}{min(a(C_i), a(\bar{C_i}))} \tag{3.4}$$

$$\phi(G) = avg(\phi(C_i)) \ , \ C_i \subseteq V \tag{3.5}$$

When a cut has high conductance, it means that many edges (or edges with high values, since conductance is also usable for weighted graphs) had to be cut to divide the graph, a situation that indicates this was a bisection of a tight group, which is undesirable. This also shows that conductance is mainly an evaluation of external cluster sparsity.

So, to better evaluate cluster quality, it is possible to define the concepts of internal and external conductance. The conductance already discussed, which mainly

evaluates the external sparsity of clusters, is the *external conductance*. The *internal conductance*, which will evaluate the internal density of a given cluster, is the conductance value of a cut that bisects a subgraph induced by the vertices of that cluster. To identify the most adequate cut to evaluate a cluster's internal density, a maximum flow minimum cut process is run. If the conductance value for the cut obtained from the max-flow min-cut is high, it means that the cluster is too dense to be adequately cut and, therefore, internally good.

3.2.4 Coverage

Coverage measures the quality of a clustering by evaluating the proportion of edges from the whole graph that connect vertices from different clusters (i.e, external edges) [Brandes et al., 2008b]. It assumes that, if only a few edges connect different clusters, than not only it means that the clusters are externally sparse, but also that those clusters are internally dense, since most of the graph's edges will be connecting vertices to other vertices in the same graph. Equation 3.6 shows how coverage is computed.

$$coverage(C) = \frac{w(C)}{w(G)}, where \tag{3.6}$$
$$w(C) = \sum_{i=1}^{k} \sum_{v_x, v_y \in C_i} w((v_x, v_y))$$

This metric gives scores between 0 to 1. Higher values mean that there are more edges inside the clusters than edges linking different clusters, which translates to a better clustering. Also, this metric can be used for weighted graphs.

3.2.5 Single cluster editing

For Single Cluster Editing (SCE), similarity is given by edge counts. It assumes that a perfect clustering structure would be formed by clusters that are completely connected internally (cliques) and completely disconnected from the rest of the graph [Shamir et al., 2004]. In order to evaluate the quality of a clustering, SCE counts the number of editions (edge insertions and deletions) would be necessary to transform this clustering into a structure similar to its conceptual "perfect clustering".

$$\epsilon_G(C_i) = \binom{|C_i|}{2} - |E(C_i)| + c_G(C_i) \tag{3.7}$$

The SCE value for a given cluster i can be obtained according to Equation 3.7, where $E(C_i) = \{(u,v)|(u,v) \in E; u,v \in C_i\}$ is the set of internal edges of cluster C_i and $c_G(C_i) = \{(u,v)|(u,v) \in E; u \in C_i, v \in V \setminus C_i\}$ is the set of intercluster edges of the same cluster. For any given graph, the higher the SCE, the worse the proposed clustering is.

3.2.6 Performance

This is another quality metric that uses edge counts as a similarity measure [Dongen, 2000]. It counts the number of edges that link vertices in a same cluster to evaluate internal cluster density. In order to evaluate a cluster's external connection sparsity, instead of counting the number of edges that connect it to the rest of the graphs, performance counts the number of edges that do *not* exist, but, if they did, would connect the given cluster to the rest of the graph. The formula used to calculate performance can be seen in Equation 3.8. This formulation assumes that the graph considered is unweighted, but there are variants for weighted graphs [Brandes et al., 2008b].

$$
\begin{aligned}
perf(C) &= \frac{f(C) + g(C)}{\frac{1}{2}n(n-1)}, where \qquad (3.8)\\
f(C) &= \sum_{i=1}^{k} |E(C_i)| \\
g(C) &= \sum_{i=1}^{k} \sum_{j>i}^{k} |\ \{\{u,v\} \notin E | u \in C_i,\ v \in C_j\}|
\end{aligned}
$$

Performance values range from 0 to 1. Higher values indicate that a cluster is both internally dense and externally sparse and, therefore, well-formed.

3.3 Semantic-Based Metrics

Sometimes data elements have extra dimensions of intrinsic information that help to describe them and their relationships. One common type of extra information comes in the form of *labels*, which are categorical descriptors which can represent intrinsic characteristics of the data. There are many metrics, such as Goodall, overlap and Eskin, that try to measure similarity between unstructured, categorical data elements [Shyam Boriah and Kumar, 2008].

Structured data represented by graphs can also have extra information in the form of categorical labels. Vertex and edge labels add an extra level of information that, while intrinsic to each element, has no direct connection with the underlying graph topology. This kind of information can be a useful tool for identifying the quality of graph clusters, and since it refers to a deeper knowledge about a graph's vertices and/or edges, metrics that use them are said to be semantic-based.

3.3.1 Entropy

Entropy is the measure of chaos (or information) in a given system. Consider the throwing of a fair coin. Since the outcome of each throw has the same probability of happening, it is very hard to predict that outcome and, therefore, each throw carries important information and thus this system has maximum entropy.

If the coin had a higher probability for one of the possible outcomes, then the result of successive throws would start to be more predictable, carrying less information and having lower entropy. Had we the situation of a coin with two heads (or tails), the outcome of a throw would always be known and, because of that, throws would carry no information and system entropy would be minimum. Entropy can be computed as shown in Equation 3.9, where $p(x_i)$ is the probability mass function of attribute x_i. It is important to note that entropy is a local metric, calculated for each cluster found.

$$E = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{3.9}$$

When applied to the evaluation of clustering quality, entropy can be used to evaluate how predictable is the distribution of labels in a given cluster. In this case, low entropy clusters will be considered better because their elements share much of their labels. This metric was used by Zhou et al. [2009] to evaluate the label-wise quality of their clustering algorithm.

3.3.2 Coherence

Gustafson et al. [2006] presented a metric to evaluate a clustering based on vertex labels. Consider, without loss of generality, that a resulting cluster of size k has x vertices with label $L1$ and y vertices with label $L2$ ($k = x + y$). To better identify how significant and/or surprising is this label distribution, the authors compare it to a random label distribution. Given the total number of vertices with labels $L1$ and $L2$

in the graph, the total number of vertices and the total size of this cluster, the authors calculate the probability that this exact proportion of labels could have occurred by chance (from a hypergeometrical distribution), for each label present in the cluster. The lowest value found will be called the p-value for that cluster.

To obtain the information on how the label distribution would occur in randomly built clusters, the authors compute the average p-value p' and the standard deviation σ_p for a large number of randomly chosen clusters. With those values, it is possible to compute the coherence index Z, as shown in Equation 3.10. Just like entropy, coherence is also a local metric, giving results about each cluster and not the whole clustering.

$$Z = -\frac{p - p'}{\sigma_p} \tag{3.10}$$

3.4 External Quality Measures

In this situation, we have an external and reliable source of knowledge that can provide the best clustering of a graph. This source usually is a specialist on the subject represented by the graph, such as a bio-engineer for a protein network graph. The problem with this solution is that it is not scalable due to its "hands-on" approach.

Another possibility is to already have an optimal or semi-optimal clustering of the desired data available, either collected with the data or based on a label that can be considered a perfect cluster identifier. In this case it is possible to use metrics like entropy, precision and revocation to evaluate how well the result from applying a clustering algorithm matches the expected result. Obviously, obtaining such optimal clustering is a big problem in itself, and this approach is good only for testing purposes, since in real situations you want to mine a graph to find the said optimal clusters exactly because they are not known.

3.4.1 Entropy

Entropy can also be used to evaluate clusterings when the true clustering is known beforehand. When used as an external quality metric for a clustering, it considers the labels of the vertex to be the real world clusters that a vertex belongs to, and evaluates the label diversity inside each cluster in the clustering [Steinbach et al., 2000b]. If one given label appears either in a lot or too very few of the vertices in one cluster, then it's

occurrence is predictable and the entropy for that cluster will be low, which indicates a good cluster.

3.4.2 F-Measure

In information retrieval, precision and recall are the most notorious metrics used to evaluate the quality of results. Precision measures the amount of correct entries in the resulting cluster, while recall indicates how many of the correct entries are represented in the resulting cluster. Since maximum recall can be guaranteed if all elements are in the same cluster and maximum precision certainly occurs if the size of the clusters is 1, and those two cases are clearly bad clustering solutions, it is necessary to balance those two quality indexes to obtain more interesting and useful results. The F-Measure aggregates those two concepts into a single, monotonic function [Steinbach et al., 2000b] and can be obtained using Equation 3.11. The F-measure gives values between 0 and 1, with higher values meaning that the clusters are more similar. Also, this measure is local, giving results for a single cluster and not the whole clustering.

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{3.11}$$

3.4.3 Rand Index

The Rand Index is a measure of the similarity between two data clusterings. This index is computed according to Equation 3.12, where a is the number of vertex pairs which are grouped in the same cluster in both clusterings, b is the number of vertex pairs which are grouped in different clusters in both clusterings and n is the number of vertices in the graph. It gives results between 0 and 1, with higher values being more desirable, and it is also a local metric.

$$R = \frac{a + b}{\binom{n}{2}} \tag{3.12}$$

Chapter 4

Evaluation of Graph Clustering Quality Metrics

In the previous Chapter, we presented some of the most popular metrics for graph clustering quality evaluation in the literature. However, if there are so many quality metrics, which one of them gives the best, most accurate results? Do they correctly score the structure of clusters?

To answer those questions, we studied some of the quality metrics previously discussed, namely modularity, silhouette, conductance, single cluster editing, coverage, and performance, in order to identify how well they perform when used to evaluate clusters in simple, undirected and unweighted graphs – the simplest kind of graph possible. We evaluated them in terms of their formulations and through experiments, so that we could better understand their results.

4.1 Motivation

Many works in the literature use quality metrics, such as the ones presented in Chapter 3, to evaluate the quality of clusterings. By discussing how some of those works use such quality metrics, we intend to show why a deeper study on the accuracy of those quality metrics popularly used in the literature is necessary.

For many works in the literature, quality metrics evaluate the similarity of a given clustering and a believed-to-exist perfect or expected clustering of a network. Since those metrics are believed to represent the "ground truth" for the network's clustering, their scores can be used to compare the efficiency of different clustering algorithms. One example of this kind of work is [Brandes et al., 2008b], which compares Markovian, iterative conductance cut and geometric MST (minimum spanning tree) clustering al-

gorithms using conductance, coverage and performance as evaluation metrics. Another example is that of Gustafson et al. [2006], where the authors use modularity and silhouette to compare K-means and hierarchical clustering techniques. [Danon et al., 2005] compare many different algorithms for graph clustering, including agglomerative, divisive and modularity maximization techniques, using modularity. Steinhaeuser and Chawla [2010] also compare different clustering algorithms using modularity, but they also use external validation metrics, since the networks used had a known expected partitioning. Other works, like those of Brandes et al. [2008a] and [Jiang et al., 2009], extrapolate the notion of using a quality metric only to evaluate results and propose algorithms that achieve clustering through modularity optimization.

Another work uses external conductance to identify the characteristics of community structures in graphs, such as the existence of a maximum or expected size for well formed clusters [Leskovec et al., 2008]. To do so, the authors propose the Network Community Profile (NCP) plots, which are graphs that present the best conductance values for clusters of different sizes. This kind of plot was used in works such as that of Gleich and Seshadhri [2012] as a tool to evaluate clustering quality. Also, in a follow-up work, Leskovec et al. [2010] extended their proposed NCP plots by using different metrics as objective functions that try to capture the classical cluster quality intuition and are popular in the field like, for example, modularity.

However, in all those discussed works, the authors simply assume that the quality metrics chosen for their experiments are good enough in the task of correctly evaluating clusters, without concerning themselves with the strength of this claim. That is problematic, as there is no consensus in the literature about the quality and effectiveness of clustering evaluation metrics. For example, Brandes et al. [2003] notice that minimum cuts have maximum coverage and, in this sense, would be considered "optimal" clusterings. However, when evaluated manually, min-cut based clusterings are not considered to be good. In another example, Gustafson et al. [2006] defend that silhouette, as a metric, is not a good enough, deciding to use only modularity in their studies. At the same time, Good et al. [2010] show that modularity maximization as a clustering strategy generates results that should be interpreted cautiously, casting great doubts on the effectiveness of modularity. Schaeffer [2007] and Kannan et al. [2004] go even further, claiming that, given the application-specific nature of clustering problems, it is probably impossible that any one quality measure can be considered universally "right".

Considering the lack of consensus about this topic, we believe that it is crucial to make a deep evaluation of the quality metrics for graph clustering already available in the literature. Our goal is to identify which one of them, if any, correctly evaluate

the structural quality of a cluster or clustering. If none of them fits this description, to gather enough information about their problems in order to be able to propose a new, better structured quality metric.

Our proposed work is similar in approach to the one by Tan et al. [2002], where the authors present a comparison between many metrics used to determine the "interestingness" of association metrics like lift, confidence and support, widely used in data mining. They show that there is no single metric that is consistently better than the others for all different scenarios and, because of that, the metrics should be chosen case-by-case to fit the expectations of the domain experts. Our work does a similar comparison for graph clustering validation metrics.

Another work that presents a similar goal to ours is the one by Shyam Boriah and Kumar [2008], which compares a large amount of similarity measures used for categorical data clustering. Yet another paper with similar goals is the one by [Abrahao et al., 2012]. In that paper, the authors seek to identify how similar the results from clustering algorithms are in comparison to the clusters obtained from external annotations made on the same network. In order to compare those cluster structures, the authors use many structural indexes, such as conductance, network diameter and betweeness, among others. This paper does something similar to what we do: evaluate the structure of the clusters themselves, instead of trying to simply compare clustering algorithms. The greatest difference between both works is their use of external information for quality evaluation. Our work uses only internal and structural information, as we believe that external information might describe clusters that are based on characteristics extrinsic to the graph structure itself and, therefore, incompatible with the purely structural evaluation we pursue.

One other common detail in many of the previously mentioned works is that the size of the networks used was rather small, with at most a couple hundred of vertices each. This poses a problem, as evaluations performed on them can hardly be assumed to remain true for larger networks, as characteristics and structures can change with the network's scale [Faloutsos, 2010]. Since automatic clustering evaluation becomes even more necessary for larger networks, as manual evaluation on those cases becomes unfeasible, ways of testing the effectiveness of algorithms on larger networks must be found. One way to do this is through the use of synthetic graphs, generated through models such as the one presented by Lancichinetti and Fortunato [2009]. That paper proposes a random graph generating model which respects complex network characteristics, such as power law degree distributions, while at the same time generating clusters that will be known *a priori*. This kind of model allows other researchers, such as Pan et al. [2010], to evaluate results with external clustering quality metrics. How-

ever, we cannot guarantee that the synthetic models for graph generation have the expressive power in order to to correctly represent all characteristics that exist in real networks. Because of that, the experiments done in this work use only graphs obtained from real world networks.

4.2 Evaluating Cluster Quality Metrics

In this section, we take a closer look at the topological clustering quality metrics discussed in Chapter 3. Our goal is to identify if those metrics' formulations correctly score clusterings in terms of the internally dense, externally sparse "ideal" cluster. The graph definitions used here are the same ones used in that chapter.

4.2.1 Modularity

Observe the classical modularity formula (originally shown in Equation 3.1):

$$Q = Tr(e) - ||e^2||$$

It gives scores based on the matrix e, which is square and has in each position e_{ij} the proportion of edges linking vertices from clusters C_i to C_j. So, it's first term, $Tr(e) = \sum_k^{i=1} e_{ii}$, mainly represents the global internal density of all clusters, as it counts all internal edges of the graph. Notice, however, that the number of vertices of each cluster is not factored in any part of this formulation, specially in its internal density term.

When evaluating external sparsity, it is not a problem to use only edge counts and ignore vertex count, as this is a relationship between two clusters and, therefore, has little to do with the vertices in either of them by themselves. However, internal cluster density does not benefit from such restrictions. Density is a measure of *concentration* and, as such, needs to represent not only the number of elements observed, but the space they occupy. In this case, it means that the concept of internal cluster density cannot be fully represented by just using the number of internal edges of a given cluster and ignoring the number of vertices of that same cluster. For example, a clustering composed by two clusters with one external edge and twelve internal edges can be represented by many different scenarios if we ignore the vertex count of each cluster. For example, it can be formed by two 4-cliques connected by one edge (definitely dense clusters), or by two 6-vertex trees connected by one edge (not dense at all). As they

were proposed, both conductance and modularity would score those two clusterings equally, and with very good scores of 0.14 and 0.34, respectively.

4.2.2 Silhouette Index

The silhouette index formula (originally shown in Equation 3.3) presents some limitations. First of all, it is very computationally expensive to calculate, requiring an all pairs shortest path execution, which is solvable in $O(V^3)$ by Floyd's algorithm.

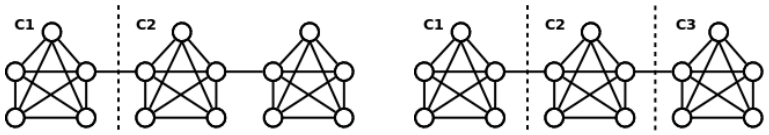
$$S(C_i) = \frac{\sum_{v \in C_i} S_v}{|C_i|} \ , \ where \ \ S_v = \frac{b_v - a_v}{max(a_v, b_v)}$$

Other problem with silhouette is how it behaves in the presence of singleton clusters. The way silhouette was proposed, since a singleton possesses no internal edges, its internal distance will be 0, which causes it to be wrongly scored as perfect. This way, clusterings with many singletons will tend to have high silhouette scores, no matter the quality of the other clusters.

4.2.3 Conductance

Even though the use of both internal and external conductances would give a most accurate assessment of both the internal density and external sparsity of a cluster, many researchers, such as Leskovec et al. [2008] and Leskovec et al. [2010], use only external conductance to evaluate cluster quality. This happens because internal conductance calculation requires the identification of minimum cuts with maximum flow for each subgraph induced by the clusters in C, a process that is computationally expensive ($O(V^3)$ with Edmonds–Karp algorithm).

Because of that, one negative characteristic of conductance, the way it is commonly used in the literature, is that it will give quality results that are biased towards clusters which are sparsely connected to others, regardless of their sizes. This means that conductance might have a tendency to give better scores to clusterings with fewer clusters, as more clusters will probably have a higher number of external edges between each cluster. Also, the lack of internal edge density information causes its evaluation results to suffer from the same problems observed for modularity, and that can be seen in Figure 4.1, where both clusterings shown would have the same conductance score, even though the one in Figure 4.1b has a clearly better structure.



(a) Two clusters. (b) Three clusters.

Figure 4.1: Two possible clusterings of a same graph.

4.2.4 Single cluster editing

SCE presents a fundamental problem when used as a quality metric: it is unbounded. This lack of bounds makes it hard to evaluate the quality of a single clustering using only its SCE value.

With that in mind, consider that $SCE(C) = \sum_{i=1}^{|C|} SCE(C_i)$. With a little arithmetical tinkering, we can obtain the following correlation:

$$SCE(C) = \sum_{C_i \in C} \binom{|C_i|}{2} - w(C_i) + |E(C_i, C \setminus C_i)|$$

$$SCE(C) = \sum_{C_i \in C} \binom{|C_i|}{2} + w(C_i) - \sum_{C_i \in C} w(C_i)$$

$$SCE(C) = \sum_{C_i \in C} \binom{|C_i|}{2} + w(C_i) - w(C)$$

$$SCE(C) = \sum_{C_i \in C} \binom{|C_i|}{2} + w(C_i) - coverage(C) \times w(G) \tag{4.1}$$

As high values of coverage and low values of SCE mean better clusters, we can see that those two quality metrics have direct correlation. Since coverage is bounded, we will ignore SCE in favor of coverage from this point on.

4.2.5 Coverage

Consider the formula for coverage (originally shown in Equation 3.6):

$$coverage(C) = \frac{w(C)}{w(G)}$$

From its formulation, we can see that the main clustering characteristic needed for a high value of coverage is inter-cluster sparsity. Internal cluster density is in no way taken into account by this metric, and it tends to cause a strong bias toward clusterings with less clusters. This can be seen in the example in Figure 4.1, where

the clustering with two clusters would receive a better score than the clearly better clustering with three clusters.

4.2.6 Performance

Consider the formulation for performance (originally shown in Equation 3.8), as shown here:

$$perf(C) = \frac{f(C) + g(C)}{\frac{1}{2}n(n-1)}, where$$

$$f(C) = \sum_{i=1}^{k} |E(C_i)|$$

$$g(C) = \sum_{i=1}^{k} \sum_{j>i} |\ \{\{u, v\} \notin E | u \in C_i,\ v \in C_j\}|$$

It is clear that performance is based on two terms: $f(C)$ counts the number of internal edges of the clusters, while $g(C)$ evaluates the external sparsity of clusters by counting the edges that would connect those clusters if they were present. The internal density evaluation term for performance, just like modularity, coverage and conductance, does not use vertex info in its formulation, and should have the same peculiarities in its behavior. However, the external sparsity term $g(C)$ carries the greater potential problem. When applied to larger networks, specially complex ones, which are sparse by nature, there is a great possibility that $g(C)$ becomes so high that it will dominate all other factors in its formula, awarding high scores indiscriminately.

4.3 Experiments

This section presents the experiments used to help evaluate the quality metrics considered. We will briefly describe our methodology and the graphs used first, and then discuss our results.

4.3.1 Methodology

We implemented the five quality metrics discussed in the previous section. To evaluate their behavior, we applied them to clusters obtained through the execution of four classical graph clustering algorithms on five large, real world graphs that will be briefly discussed in the next subsection. This variety of clustering algorithms and graphs

is necessary to minimize the pollution of the results by possible correlations between metrics, algorithms, and/or graph structures.

We used freely available implementations for all clustering algorithms: the *MCL* implementation by Van Dongen, which is available with many Linux distributions in a package of the same name, the implementation of bisecting K-means available in the Cluto suite of clustering algorithms [1], the spectral clustering algorithm implementation available in SCPS [Nepusz et al., 2010] and the normalized cut clustering implementation GRACLUS [Dhillon et al., 2007].

Three different inflation indexes were chosen for the MCL algorithm, based on the values suggested by that algorithm's documentation: 1.5, 2, and 3. The number of clusters found using each MCL configuration were used as the input for the other algorithms, so that we could compare clusterings with roughly the same number of clusters.

4.3.1.1 Graphs

We used seven different datasets derived from real complex networks. Two of them are smaller, with known expected partitions that could be used for comparison, and the other five are bigger, with no known expected partitions. All graphs used are undirected and unweighted.

The first small dataset is the Karate club network. It was first presented by Zachary [1977] and depicts the relationships between the students in a karate dojo. During Zachary's study, a fight between two teachers caused a division of the dojo in two, with the students more related to one teacher moving to his new dojo. Even though this dataset is small (34 vertices), it is interesting to consider because it possesses information about the real social partition of the graph, providing a ground truth for the clustering.

The other small dataset used was the American College football team's matches described by Girvan and Newman [2002], composed of 115 vertices and 616 edges. It represents a graph where the vertices are football teams and an edge links two teams if they played against each together. Since the teams play mostly with other teams in the same league, with the exception of some military school teams, which belong to no league and can play against anyone, there is also an expected clustering already known for this graph.

The five remaining networks were obtained from the Stanford Large Network

[1] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

Dataset Collection[2]. Two of them represent the network of collaborations in papers submitted to the arXiv e-prints in two different areas of study, namely Astrophysics and High Energy Physics. In those networks, researchers are the vertices, and they are linked by edges if they collaborated in at least one paper. The Astrophysics network is composed by 18,772 vertices and 396,160 edges, while the High Energy Physics has 12,008 vertices and 237,010 edges. Another network based on the papers submitted to the arXiv e-prints was used, but covering the citation network of authors in the High Energy Physics category. In this case, an edge links two authors if one cites the other. This network has 34,546 vertices and 421,578 edges.

The last two networks are snapshots from a Gnutella peer-to-peer (P2P) file sharing network, taken in two different dates. Here the vertices are the Gnutella clients and the edges represent the overlay network connections between them. The first snapshot was collected in August 4, 2002 and contains 10,876 vertices and 39,994 edges. The second one was collected in August 30, 2002 and has 36,682 vertices and 88,328 edges.

4.3.2 Results

We first considered the smaller datasets, Karate Club and College Football, to check how the algorithms and quality metrics behave in small networks where the expected result was already known. The results for the Karate Club dataset can be seen in Table 4.1. The College Football dataset had similar results and was omitted for brevity. The results shown represent the case with two clusters, which is the expected number for this dataset. It can be observed that the scores obtained were fairly high. Also, the resulting clusters were very similar to the expected ones, with variations of two or three wrongly clustered vertices. However, those two study cases were very small and classical, so good results here were more than expected, as most of the quality metric biases we pointed out in the beginning of this chapter were expected for bigger networks with many clusters.

Algorithm	SI	Mod	Cov	Perf	Cond
MCL	0.13 ± 0.02	0.29	0.71	0.55	0.55 ± 0.15
B. k-means	0.081 ± 0.001	0.37	0.87	0.62	0.26 ± 0.13
Spectral	0.13 ± 0.02	0.36	0.87	0.61	0.30 ± 0.15
Norm. Cut	0.14 ± 0.017	0.18	0.68	0.56	0.65 ± 0.32

Table 4.1: Karate Club dataset and its quality indexes for two clusters.

[2]http://snap.stanford.edu/data/

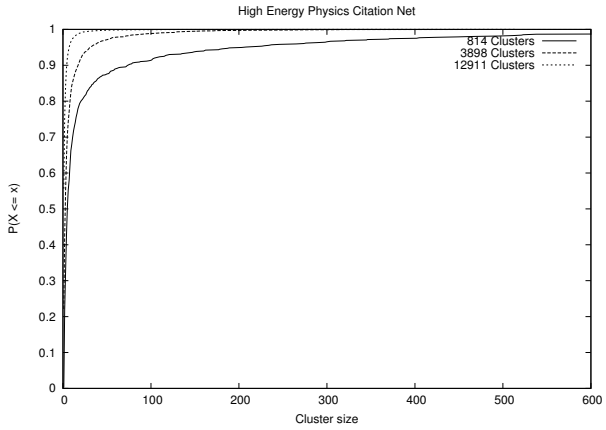
Now, for the larger datasets. The quality metric values for the Astrophysics Collaboration network are shown in Table 4.2. It's already possible to observe some trends on the quality metrics' behavior, no matter what clustering algorithm is used. For example, modularity, coverage and conductance always give better results for smaller numbers of clusters. Also, we can see that, as expected from our observations, performance values have no discriminating power to compare any of our results. The silhouette index presents a somewhat erratic behavior in this case, without a clear tendency of better or worse results for more or less clusters.

Algorithm	# Clusters	SI	Mod.	Cover.	Perf.	Cond.
MCL	1036	-0.22 ± 0.038	0.35	0.42	0.99	0.55 ± 0.02
MCL	2231	-0.23 ± 0.026	0.28	0.31	0.99	0.70 ± 0.006
MCL	4093	0.06 ± 0.015	0.19	0.27	0.99	0.82 ± 0.003
B. k-means	1037	-0.73 ± 0.017	0.25	0.28	0.99	0.70 ± 0.002
B. k-means	2232	-0.48 ± 0.005	0.21	0.24	0.99	0.70 ± 0.002
B. k-means	4094	-0.21 ± 0.01	0.17	0.19	0.99	0.76 ± 0.001
Spectral	1034	-0.15 ± 0.036	0.34	0.38	0.99	0.53 ± 0.015
Spectral	2131	-0.26 ± 0.027	0.25	0.28	0.99	0.66 ± 0.007
Spectral	3335	0.04 ± 0.017	0.19	0.21	0.99	0.78 ± 0.004
Norm. Cut	1037	-0.69 ± 0.021	0.23	0.25	0.99	0.66 ± 0.006
Norm. Cut	2232	-0.51 ± 0.019	0.17	0.19	0.99	0.73 ± 0.015
Norm. Cut	4094	-0.31 ± 0.006	0.13	0.15	0.99	0.81 ± 0.0004

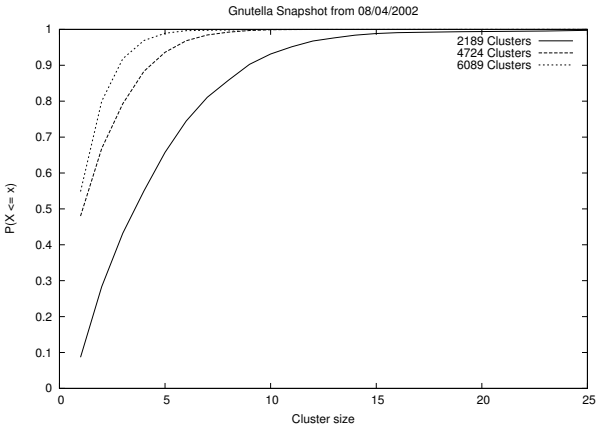
Table 4.2: Astrophysics collaboration network clusters and their quality indexes.

For the High Energy Physics Collaboration network, as we can see in Table 4.3, the tendencies observed in the previous network are still true. Also, silhouette index shows a more pronounced bias toward larger numbers of clusters. If we look at the cumulative distribution function (CDF) of cluster sizes (as shown in Figure 4.2 for just two instances of our experiments, but which are consistent with the rest of the obtained results), we can see that bigger clusterings tend to have a larger number of smaller clusters. So, this bias of the silhouette index is expected from our observations in this chapter. Those same tendencies occur in the High Energy Physics Citation network, as seen in Table 4.4.

The quality metric scores for one of the Gnutella snapshot networks can be seen in Table 4.5. The scores for the other one were very similar, so we suppressed them for brevity. It is possible to notice that the results for those graphs still present the same tendencies shown in the other cases, but with a key difference: while silhouette and performance results show no big difference from the other datasets, as they are easily fooled by high numbers of singleton clusters and network size, respectively, modularity,



(a) High Energy Physics Citation



(b) Gnutella Snapshot 08/04/2002

Figure 4.2: Some cluster size's Cumulative Distribution Functions (bisecting k-means).

coverage and conductance give much lower quality results. This happens because the structure of a Gnutella network, where common peers connect only to "superpeers", and those superpeers also connect to each other. This structure leads to a very low occurrence probability for 3-cliques (0.5% in the Gnutella networks against 31.8% in the Astrophysics Collaboration network, for example). Also, the Gnutella networks presented here are way sparser than the other studied networks, with only 6.76% of all possible edges present in the graph for the August 04 snapshot, against 32.88% for the High Energy Physics citation network, for example.

4.3.2.1 Discussion

For all the generated cases, coverage, modularity and conductance have better values for smaller numbers of clusters. This behavior is expected from the formulation of coverage, since it observes the number of inter-cluster edges, which tends to be smaller if there are less clusters to link to. The same thing happens to conductance, as more inter-cluster edges mean more expensive cuts. Without balancing the external conductance with the internal conductance, results will only give us partial and biased results.

Concerning modularity, we already know that singleton clusters have a very bad impact on the modularity score, and the larger the number of clusters, the bigger the chance of singletons occurring. It is interesting to notice that giving low scores to singleton clusters is not wrong *per se*, but since those scores influence the overall score of the clustering, they can obfuscate the existence of well scored clusters in the final tally.

Silhouette Index generally gives better results for more clusters, which can also

be attributed to the larger occurrence of singletons, which wrongly get optimal results for SI.

Algorithm	# Clusters	SI	Mod	Cov	Perf	Cond
MCL	1002	-0.17 ± 0.037	0.35	0.52	0.99	0.51 ± 0.016
MCL	1742	-0.17 ± 0.028	0.33	0.42	0.99	0.62 ± 0.009
MCL	2650	0.005 ± 0.019	0.22	0.27	0.99	0.73 ± 0.005
B. k-means	1005	-0.54 ± 0.012	0.33	0.41	0.99	0.61 ± 0.007
B. k-means	1744	-0.30 ± 0.004	0.30	0.37	0.99	0.61 ± 0.006
B. k-means	2652	-0.14 ± 0.016	0.25	0.31	0.99	0.68 ± 0.003
Spectral	1005	-0.16 ± 0.037	0.34	0.44	0.99	0.53 ± 0.015
Spectral	1710	-0.04 ± 0.025	0.29	0.35	0.99	0.64 ± 0.009
Spectral	2525	0.019 ± 0.019	0.25	0.29	0.99	0.71 ± 0.006
Norm. Cut	1005	-0.59 ± 0.025	0.26	0.33	0.99	0.64 ± 0.02
Norm. Cut	1744	-0.37 ± 0.01	0.18	0.21	0.99	0.70 ± 0.01
Norm. Cut	2652	-0.25 ± 0.014	0.18	0.23	0.99	0.76 ± 0.015

Table 4.3: High energy physics collaboration network clusters and their quality indexes.

For performance, as we already expected from the discussion about the formula itself, the sheer size of the networks we worked with eclipsed any kind of meaningful results we could gather from the clusterings themselves. The results here serve as a confirmation that the predicted behavior really happens on real networks.

Algorithm	# Clusters	SI	Mod	Cov	Perf	Cond
MCL	814	-0.07 ± 0.037	0.41	0.43	0.98	0.58 ± 0.015
MCL	3898	-0.039 ± 0.017	0.26	0.26	0.99	0.81 ± 0.003
MCL	12911	0.41 ± 0.005	0.12	0.12	0.99	0.93 ± 0.0006
B. k-means	814	-0.71 ± 0.014	0.25	0.25	0.99	0.71 ± 0.005
B. k-means	3898	-0.64 ± 0.008	0.14	0.14	0.99	0.80 ± 0.004
B. k-means	12911	-0.077 ± 0.01	0.06	0.056	0.99	0.90 ± 0.0008
Spectral	812	-0.236 ± 0.04	0.34	0.35	0.99	0.59 ± 0.014
Spectral	3490	0.043 ± 0.016	0.20	0.21	0.99	0.81 ± 0.003
Norm. Cut	814	-0.74 ± 0.006	0.25	0.25	0.99	0.65 ± 0.003
Norm. Cut	3898	-0.70 ± 0.005	0.10	0.10	0.99	0.82 ± 0.002
Norm. Cut	12845	-0.004 ± 0.006	0.06	0.06	0.99	0.92 ± 0.0006

Table 4.4: High energy physics citation network clusters and their quality indexes.

Another important point raised by our experiments is that networks of different origins might have clusters with very different characteristics. Clusters obtained from technological networks (in our case, the Gnutella snapshots) got markedly poor quality

metric results, specially when compared to the results from social networks (all the other networks used). It could be argued that those technological networks in particular might not have clusters, but we know that there should be community-like structures in a Gnutella network: a superpeer and its neighboring peers form a fairly cohesive subset, even though it may be considered sparse when compared to a social network cluster.

Algorithm	# Clusters	SI	Mod	Cov	Perf	Cond
MCL	2189	-0.81 ± 0.039	0.0004	0.001	0.99	0.99 ± 0.0
MCL	4724	-0.037 ± 0.015	0.0003	0.0007	0.99	0.99 ± 0.0
MCL	6089	0.10 ± 0.011	0.00003	0.0003	0.99	1.00 ± 0.0
B. k-means	2189	-0.88 ± 0.0001	0.0004	0.001	0.99	0.99 ± 0.00034
B. k-means	4724	-0.52 ± 0.02	0.00007	0.0004	0.99	0.99 ± 0.0
B. k-means	6089	-0.18 ± 0.01	-0.00006	0.0002	0.99	1.00 ± 0.0
Spectral	2158	-0.90 ± 0.0006	0.0004	0.001	0.99	0.99 ± 0.0
Spectral	4079	-0.94 ± 0.0005	0.0001	0.0005	0.99	0.99 ± 0.0
Spectral	6089	-0.30 ± 0.02	-0.00007	0.0002	0.99	1.00 ± 0.0
Norm. Cut	2189	-0.90 ± 0.002	0.0003	0.001	0.99	0.99 ± 0.0
Norm. Cut	4616	-0.2 ± 0.012	0.00025	0.0006	0.99	0.99 ± 0.0
Norm. Cut	5690	0.1 ± 0.012	0.0002	0.0005	0.99	0.99 ± 0.0

Table 4.5: Gnutella peers network (08/04/2002) clusters and their quality indexes.

It seems that the network structure in this case, with its non clique-like communities, affects very negatively the ability of both clustering algorithms and quality metrics to identify any clusters. This observation, that different kinds of cluster structures exist for technological networks and that the usual clustering methods wouldn't work with them, was already discussed by Nepusz and Bazso [2007]. In that case, the authors defended that, in a bipartite graph, each side of the bipartition should be considered as a cluster. Kumar et al. [1999] also mention the existence of this kind of cluster structure, pointing out that there are many on-line communities that behave as bipartite subgraphs, offering the websites of cellphone carriers as an example: they represent the same category of service, but will not have direct links to each other.

Chapter 5

Proposal for a New Quality Metric

In the previous chapter, we saw that the graph clustering evaluation metrics most commonly used in the literature present strong biases, and those biases raise doubts about the quality of their results. So, there is a strong need to find better ways to evaluate cluster quality, structurally-wise. In this chapter, we will identify the most critical problems not correctly addressed by the quality metrics studied so far and, based on those observations, propose a new evaluation metric.

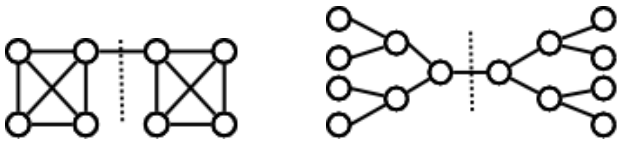
5.1 Problems of Current Quality Metrics

From our studies and experiments, shown in Chapter 4, we were able to identify two main problems in the currently used structural quality metrics. The first one is the way those metrics evaluate internal cluster density, which is one of the two main characteristics classically associated to structurally good clusters. The other problem is the fact that the studied quality metrics evaluate networks from different origins by the same standards. Those two problems will be better discussed now.

5.1.1 Incorrect Internal Density Evaluation

Observing the formulation of both modularity and conductance (Sections 3.2.1 and 3.2.3), for example, it is possible to see something in common — both metrics use only edge counts, ignoring vertex counts, in order to infer the internal density and external sparsity of a cluster. Modularity uses the matrix e, which counts edges connecting elements inside and outside each cluster as the basis of its formulation, while conductance uses the ratio between edges connecting elements from the evaluated cluster to other clusters and all edges with at least one endpoint in the evaluated cluster.

When evaluating external sparsity, it is not a problem to use only edge counts, as this is a relationship between two clusters and, therefore, has little to do with its vertices in particular.



(a) Two dense clusters.

(b) Two sparse clusters.

Figure 5.1: Very different clusterings that are equally good for most of the currently used quality metrics evaluated.

However, internal cluster density has strong dependency on the vertices of the studied cluster. Density is a measure of *concentration* and, as such, needs to represent not only the number of elements observed, but the space they occupy. In this case, it means that the concept of internal cluster density cannot be fully represented by using just the number of internal edges of a given cluster and ignoring the number of vertices of that cluster. For example, a clustering with 2 clusters composed of 1 external edge and 12 internal edges can represent many different scenarios if we ignore the vertex count of each cluster, like two 4-cliques connected by one edge (Figure 5.1a, definitely dense clusters) or two 6 vertex trees connected by one edge (Figure 5.1b, not dense at all). As they were proposed, both conductance and modularity would consider those two clusters to be equivalent, and with relatively good scores of 0.14 (conductance) and 0.34 (modularity).

5.1.2 Evaluation of Different Types of Networks by the Same Standards

Not all networks are made equal. Some might represent some kind of technological infrastructure, like the airport network formed by flight connections, or computers connected via network cables and switches, for example. In those cases, creating each connection has a real world monetary cost, and that tends to generate sparser, more deliberate edge structures. Other networks can represent social relationships, such as friendship networks on websites like Facebook or Twitter, where the cost of creating new connections between users is negligible. Therefore, networks of that kind tend to be denser.

So, to assume that a cluster from a technological network that has the same internal density as one from a social network should be similarly scored can be unnecessarily unfair to the naturally sparser network. Nevertheless, that's what quality metrics such as conductance and modularity, among others, do.

The edge count-based scoring method adopted by those quality metrics causes their results to be always lower, i.e., worse, for graphs that are naturally sparser, a behavior that can be seen in the results presented in Chapter 4 and in works such as Zaidi et al. [2010]. Because of that, it's difficult to compare how good a clustering algorithm is for graphs of different origins, as the metric results will be mostly incomparable.

This raises an interesting point: density is a malleable concept, since what is considered dense for a given network is not guaranteed to be considered so for other, different networks. So, we believe that it is important to find ways to set the threshold that defines what is dense and what is not, regarding any given network.

5.2 Proposal for a New Quality Metric

Based on all the information gathered so far, we propose a new quality metric that takes into account the two main problems identified and discussed in the previous section. This metric structurally evaluates simple, undirected and unweighted graphs, using only internal graph information. It is composed by two separate components, as works in the literature suggest that bi-criteria measures are empirically observed to obtain better results for cluster evaluation [Kannan et al., 2004]. One of the components of our metric evaluates the internal density of each cluster, while the other evaluates the external sparsity among them. Those components are combined to form a unique metric, although taking the component values separately might also be useful to better understand cluster structures.

5.2.1 Internal Density Component

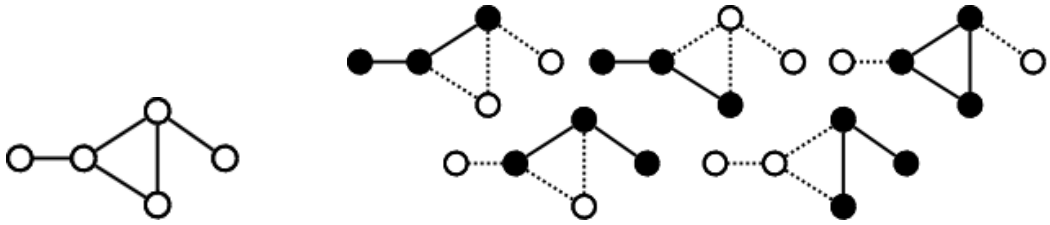
According to Newman and Girvan [2004], the internal density of a cluster is one of its two main structural characteristics. However, as discussed in Section 5.1, the currently used quality metrics do not evaluate this characteristic correctly. So, in order to more effectively evaluate a cluster's internal density, our proposed metric takes the two problems discussed in that section into consideration.

To solve the incorrect evaluation of internal density, vertex count information should be considered during cluster evaluation. One possible solution for this could be

to compare only subgraphs of the same size (vertex-wise). In that case, edge counts can be correctly considered as density indicators.

As for the inherent difference in clusters structures for different kinds of networks, the quality metric must use information derived from the target graph to identify the thresholds that indicate the density/sparsity of the clusters found. Doing so will allow our method to positively score clusters that can be considered dense for a given graph, even if they would not be considered dense in other graphs.

So, considering one cluster with s vertices from a given graph, we want to be able to identify how its internal edge count compares to what is the expected from clusters of size s in that same graph. If the evaluated cluster has more edges than expected, than it should be scored accordingly. By doing that, we can evaluate a cluster in terms of what is expected based on that specific graph, while also incorporating the number of vertices of the cluster to the evaluation process, addressing the two problems discussed previously.



(a) Original graph. (b) Population of size 3 subgraphs.

Figure 5.2: A simple graph and its subgraphs.

One way to discover the expected internal edge count for a cluster of size s is to consider the whole population of all possible connected induced subgraphs of that same size in the network at hand. For example, consider that we want to define the density thresholds for a size 3 cluster obtained from the graph in Figure 5.2a. If we enumerate all possible size 3 induced connected subgraphs from this graph (as shown in Figure 5.2b), we can see that most of them have only 2 edges, while only one subgraph has 3 edges. Since a 3-edged subgraph is more edge-dense than 80% of all possible subgraphs of that same size from that particular network, then a size 3 cluster with 3 internal edges found in this same graph should be positively scored based on this information.

However, the example in Figure 5.2 is small and very simple. As the graph grows in size, the population of possible connected subgraphs of a given size becomes so large as to be intractable to enumerate. So, it is necessary to use a sample from that population in order to use that approach.

5.2.1.1 Sampling for Density Evaluation

Most works in the literature use cluster quality metrics to evaluate results from traditional graph clustering algorithms. Some examples of this kind of work are [Brandes et al., 2008b], [Du et al., 2007], [Brandes et al., 2003] and [Leskovec et al., 2008]. In all those cases, the universe of evaluated clusters is limited to what the clustering algorithms used identify as a (good) cluster. This kind of filtering may introduce biases on what kinds of structures will be evaluated and, therefore, may bias the final quality evaluation results.

To avoid this kind of bias in our evaluation method, we propose to use samples from the universe of all possible connected induced subgraphs with s vertices that exist in a given graph. By doing so, our results will represent, within statistical guarantees, the expected internal density values for a size s cluster found in a given graph, and not just the ones that are deemed to be "interesting" by a given clustering algorithm. To simplify the discussion here, we leave the description of a sampling process that provides such guarantees for Section 5.2.1.2.

Using this technique, Figure 5.3 shows the number of sampled 25-vertex subgraphs by edge count for three complex networks of different origins: Amazon Co-purchases, Yeast protein interactions and Google websites (social, biological and technological, respectively). Those networks will be better described in Section 6.1.2.

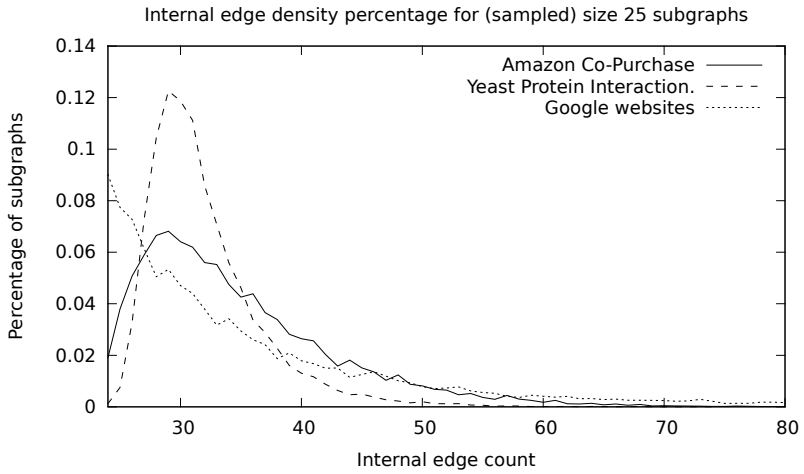


Figure 5.3: Example of internal densities in subgraph samples ($s = 25$).

Since we are looking at induced subgraphs with a fixed vertex count, it is valid to equate edge count and density in this case. It is possible to see that those density curves reach their peaks very near the minimum possible density ($|s| - 1$ edges, since the subgraphs are connected), and that they are somewhat heavy-tailed. That means

that not only most of the sampled subgraphs of a given size are way too sparse to be considered good clusters, but also that there are a few subgraphs that are denser than what is expected from the average case. We believe that it is possible to use the information presented in this kind of graph to evaluate cluster quality in a way that bears more significance to the graph considered.

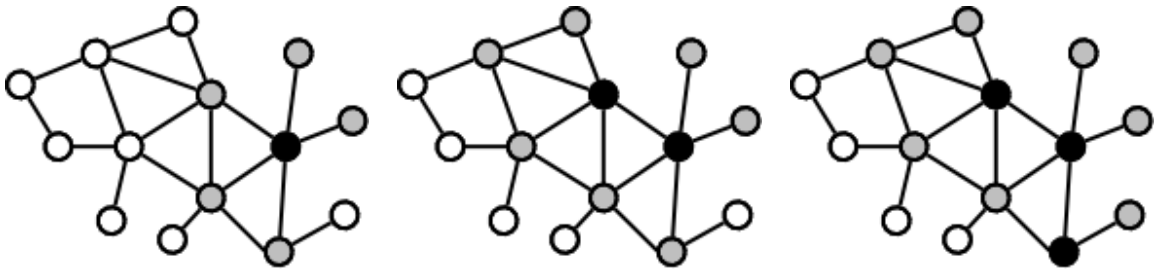
5.2.1.2 Sampling Process

Given a graph $G = (V, E)$, we want to identify the expected edge count for a connected induced subgraph with s vertices. Also, we want to do it without having to enumerate all possible subgraphs, as this can be unfeasible for larger graphs.

One way to do that is through sampling of the universe of all possible s-sized connected induced subgraphs from G. In order to obtain an estimate of the proportion of edge densities in that universe, with a confidence level of 99% and margin of error of 1%, we can perform a simple random sampling without replacement of at least 2477 subgraphs from that universe, a value obtained through Equation 5.1 [Lohr, 2010; Berman, 2012], which assumes that the universe in question is of unknown size, but is known to be considerably large. In this equation, $Z = 1 - \frac{\alpha}{2}$ (α is the confidence level desired), ME is the margin of error desired and p is the probability for one sampled element to have a given characteristic. Variable p was set with the suggested (conservative) value of 0.5, which guarantees that a sample of the calculated size would have *at least* the level of confidence and *at most* the measurement error chosen. We were even more conservative and used samples of 8000 subgraphs, more than enough to ensure the measuring error and confidence level desired.

$$n = \frac{[Z^2 \times p \times (1-p)] + ME^2}{ME^2} \tag{5.1}$$

The process of choosing one random connected subgraph from a given universe is done as follows: we pick one vertex randomly from all vertices in the graph (Figure 5.4a). We pick the next vertex randomly from the immediate neighborhood of that vertex (Figure 5.4b). For the third and following vertices, we pick randomly one of the vertices from the set of the immediate neighbors to any of the vertices already chosen (Figure 5.4c). It is important to note that if one vertex is in the neighborhood of more than one vertex already chosen, this does not make it more likely to be picked. This process aims to be as random as possible in order to emulate the process of randomly picking one element from the universe of all s-sized connected induced subgraphs of G.



(a) One vertex chosen. (b) Two vertices chosen. (c) Three vertices chosen

Figure 5.4: Example of the sampling process. Black vertices were chosen and gray vertices are the extended neighborhood.

5.2.1.3 Difference Between Subgraph Sampling and Graph Sampling

It is important to notice that what we are doing is different from the problem of graph sampling. We use a number of samples from the set of all possible subgraphs of the same size to estimate the internal edge density of that set, based on the samples. Graph sampling, as discussed by Leskovec and Faloutsos [2006] , Ribeiro and Towsley [2010] and Clauset and Moore [2005], aims to discover subgraphs that present the same characteristics (such as degree distribution, clustering coefficient and diameter, for example) as the original graph, only in a smaller scale. Such kind of sample is used when the original graph is too large to be analyzed.

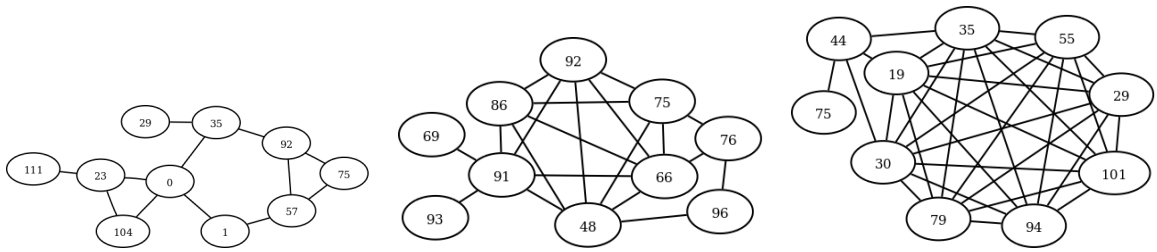
Graph sampling adds many restrictions to the sampling process, making simple random sampling inadequate to obtain a sample that can be considered valid, as can be seen in the works mentioned. Fortunately, our goal is not as complex as graph sampling. For example, consider that the universe we want to sample is a cake with filling and frosting. For a simple random sampling, one could cut the cake into small cubes and randomly pick some of them. With a large enough sample, it would be possible to identify the proportion of each one of the components on the original cake. This kind of information is good enough for our problem, but would be inadequate for the problem of graph sampling, as it would not be possible to infer the structure of the cake, i.e., you could not guarantee that the frosting was indeed covering the cake. For graph sampling, a slice would be a better sample.

5.2.1.4 Internal Density Index

With those internal density distributions in mind, we should be able to define a model that implements the proposed method for internal density evaluation, which will be used as the internal quality component of our quality metric.

For this model, to evaluate the density of a given cluster of size s and e edges from a given graph, we will use the expected number of edges for a subgraph of the same size in the same graph. To do so, we could use the percentile of value e in the internal density curve obtained from the sampling of size s clusters in the graph as the quality score for that cluster. This simple model would allow better scores for larger internal density values, with the 50^{th} percentile of the distribution being assumed to be a "neutral" score. Higher percentiles would mean better scores, and lower percentiles, worse ones.

However, this schema would have poor discerning power to compare internal density values in the tail of the curve. For example, Figure 5.5 presents 3 different clusters of the same size, but with different densities, obtained from the college football graph. If we use this simple percentile model, than the cluster in Figure 5.5a would get a score of 0.2598, the one from Figure 5.5b would score 0.9623 and the one from Figure 5.5c would score 0.9999. The first cluster, which is almost as sparse as a tree, presented a low score, as expected. The two other graphs received high, but relatively close, scores. The problem here is that the third cluster has more than 50% more edges than the second one, and their scores should better reflect that disparity.



(a) 12 Edges (b) 20 Edges (c) 32 Edges

Figure 5.5: Example of sampled subgraphs of size $s = 10$ from the College Football graph.

One way to improve this model would be to create a two-phase scoring system. In such a system, we could shift the percentile value that would score as the "neutral" value (0.5) to a higher percentile, so that the first scoring phase (lower than the shifted percentile) would cover a wider range of sparse density values, leaving a smaller range of possible density values to be scored in the second phase of the model.

$$IDI = \begin{cases} \frac{P(X<x)}{2D} & if\ [P(X<x)] < D \\ \\ \frac{[P(X<x)]-2D+1}{2D-2} & if [P(X<x)] \geq D \end{cases} \tag{5.2}$$

This simple model is presented in Equation 5.2. It has a parameter D, which represents the percentile that will mark the threshold which will be scored as "neutral". For example, if we assumed a value of $D = 0.75$, it would mean that the 75^{th} percentile would score 0.5 in our internal density index (IDI). This shift allows for an amplified discriminative power when evaluating the density values that matter the most: the ones in the tail of the density curve. One illustration of how this scoring method would work for $D = 0.75$ can be seen in Figure 5.6, with the values of the IDI rising slowly until the defined percentile and faster afterwards.

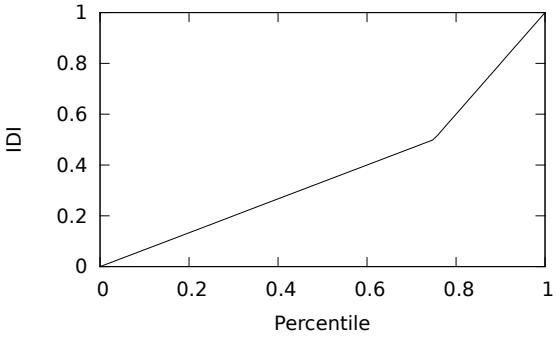


Figure 5.6: Representation of the IDI's scoring method ($D = 0.75$).

For the same example in Figure 5.5, the first subgraph would now score 0.1732, the second 0.9246 and the third, 0.9998. This simple adjustment penalized even further the sparser subgraph, while accentuating (even if slightly) the difference in density of the other two.

5.2.2 External Sparsity Component

The second component of our metric deals with the evaluation of the level of separation between clusters. From what we have seen so far in our studies (Chapter 4), even though (external) conductance does not correctly evaluate internal cluster density, it is effective in the evaluation of the external sparsity between clusters. Considering this, we decided to use it as our metrics external component.

$$\phi(C_i) = \frac{\sum_{u \in C_i} \sum_{v \notin C_i} w(\{u, v\})}{a(C_i)} \tag{5.3}$$

In our formulation, conductance will be calculated based the formula presented on Equation 5.3, which is a slightly simplified version of the one presented on Equation 3.5 in Chapter 3. In this new formulation, instead of using the minimum between

$a(C_i), a(\bar{C}_i)$ as the denominator of the equation, we use only $a(C_i)$. This is done because we want, as much as possible, to evaluate each cluster only in terms of itself, and with this simplification we will not need to evaluate any other clusters in order to calculate a given cluster's conductance score.

5.2.3 Complete Metric

In order to create a metric that can provide a more accurate indicator regarding a given cluster's structural quality, it is necessary to aggregate the internal and external density metrics discussed. To do that, we propose a simple, yet effective formulation, presented on Equation 5.4:

$$Q(C_i) = \frac{IDI(C_i) + (1 - \phi(C_i)}{2}) \tag{5.4}$$

This quality formulation will consider both the internal and external quality indexes equally when evaluating a cluster's quality. The complement of conductance $(1 - \phi(C_i))$ is used so that both indexes present the same scoring scheme, with larger values being better than lower ones. Considering this scoring method, it is fair to assume that clusters with quality value of 0.5 are either very good internally or externally, but not both. So, that score indicates clusters that are, at best, neutral, but never globally good. Clusters with quality scores higher than 0.5 are considered increasingly better, while clusters with lower scores are worse.

However, if the user wishes to consider the impact of internal and external cluster quality to be different from one another, then a simple weighting scheme can be used. Equation 5.5 presents such a scheme:

$$Q(C_i) = \alpha \times IDI(C_i) + (1 - \alpha) \times (1 - \phi(C_i)) \tag{5.5}$$

In this alternative formulation, a parameter $0 \leq \alpha \leq 1$ defines the importance of each of the two components in the final score. With $\alpha = 0.5$, for example, both internal density, represented by the Internal Density Index (IDI), and external sparsity, represented by the conductance (ϕ), are considered to be equally important for the evaluation of cluster quality, and this formulation will yield the same results as the one in Equation 5.4.

The next chapter presents an evaluation of the quality metric proposed here. This evaluation will take into consideration the metric's scoring behavior and also how it compares against other metrics from the literature.

Chapter 6

Experimental Evaluation

This chapter presents the experiments done in order to compare the performance of our proposed metric to some of the most popular quality metrics used in the literature. First, the methodology used will be presented. Later, some of the most interesting results will be discussed.

6.1 Methodology

The motivation of these experiments is to study the behavior of our metric when applied to clusters in complex networks of different origins. The following subsections will discuss the process used to obtain the clusters, the complex networks evaluated and Kendall's Tau coefficient, which was used to compare the different metrics.

Each one of the clusters obtained was scored using modularity, silhouette, conductance and our metric. Since our metric gives fine-grained, by-cluster quality evaluation scores, the modularity function used here is the one from Equation 3.2 (Chapter 3), as it makes it easier to evaluate the contribution from each cluster to the global modularity score. So, when we discuss modularity values for individual clusters, we mean the contribution from that given cluster to the modularity score for the full clustering.

The proposed metric has a couple of parameters, which were set as follows. The sample size was set to 8000 subgraphs, which is more than enough to cover the stipulated minimum of 2477 defined in Chapter 5, Section 5.2.1.2. The density cutoff D, which separates the density values which are considered interesting by the Internal Density Index was set to 0.75, a value chosen by empiric observation of the behavior of the internal density of sampled subsets of different kinds of networks. Our quality metric is the one presented in Equation 5.4, which is equivalent to Equation 5.5 with $\alpha = 0.5$ (both equations can be found in Chapter 5). In our discussions we present

results for both the full metric and its components, making it easier to assess their impact in the quality scores obtained .

6.1.1 Clustering Algorithms

The clustering algorithm implementations used in this experiment are the same ones used in the experiment discussed in Chapter 4: MCL (Markovian), Cluto (bisecting k-means), SCPS (spectral) and Graclus (normalized cut). MCL was ran for three different values of inflation index, namely 1.4, 2 and 4, in order to obtain clusterings of different levels of granularity. Those values are among the ones suggested by MCL's author as good starting values. The other three algorithms assume that the number of clusters to be found is an input parameter. So, for each graph to be clustered, the number of clusters found by MCL with the chosen inflation indexes were used as input parameters. This procedure is the same one used for the experiment shown in Chapter 4. Also, it is important to notice that not all algorithms could be run for all different algorithms and parameters due to time/memory limitations.

6.1.2 Graphs used

The datasets used in our evaluation, which can be seen in Table 6.1, can be roughly grouped in 3 different kinds of complex networks: social, biological and infrastructure (or technological).

Type	**Network**	**# Vertices**	**# Edges**
Social	Les Miserables	77	254
	Slashdot (11/2008)	77360	905468
	General Relativity Collab.	5242	28980
	Condensed Matter Collab.	23133	186936
Technological	College Football	115	616
	Power Grid	4941	6594
	Gnutella Snap. (31/08/02)	62586	147892
	Gnutella Snap. (30/08/02)	36682	88328
Biological	C. Elegans Neural Net.	297	2359
	Yeast	2361	7182

Table 6.1: Datasets studied.

In social networks, edges represent relationships between humans (or other social animals). For our experiments, we used the graph of character interactions in the novel *Les Miserables*, by Victor Hugo [Knuth, 1993]; the graph generated from the

discussions between Slashdot users, a popular tech-related news site; and the scientific collaboration graphs of researchers from General Relativity and Condensed Matter areas [Leskovec, 2012].

As for biological networks, edges represent connections that happened between biological entities. Examples used here are the protein-protein interaction network in yeast [Bu et al., 2003] and the C. Elegans neural network [Watts and Strogatz, 1998a].

The edges in an infrastructure or technological network connect its vertices through an algorithmically defined process. Also, each edge has a real world cost, making them generally sparser then other kinds of complex networks. Examples of such networks evaluated here are the topology of the Western States Power Grid of the United States [Watts and Strogatz, 1998a]; the network of American football games between Division I-A colleges during regular Fall 2000 season [Girvan and Newman, 2002]; and two snapshots, took at different times, of the connections between clients of the Gnutella peer-to-peer network [Leskovec, 2012].

6.1.3 Kendall's Tau Correlation Index

The scores obtained through the quality metrics evaluated cannot be directly compared to each other, as they do not have the same bounds nor have consistent scoring policies. Therefore, the solutions must be compared based on their relative values for each metric. Based on that, we ranked clusterings using their value for each metric in decreasing order of quality. The differences in the behavior of the multiple metrics can lead to different rankings for a given set of clusterings with different metrics. To evaluate if two metrics are consistent (if they produce the same ordering), we compare the resulting ranks (with k positions) in pairs using Kendall's Tau ranking correlation coefficient [J., 1980], given by Equation 6.1. We say that a pair is concordant for two metrics when the two objects appear in the same order in the rankings produced by the two metrics; otherwise, they are discordant.

$$\tau = \frac{(\#concordant\,pairs) - (\#discordant\,pairs)}{\frac{1}{2}n(n-1)} \tag{6.1}$$

The *Kendall's Tau metric* gives values between -1 and 1, with -1 representing a total dissimilarity between the rankings, and 1, a perfect match between them. This metric provided us with a way to observe how the quality measures compare to each other in terms of the rankings they yield.

6.2 Results

In this section, we will use the results obtained from the previously described experiments to discuss the scoring behavior of our quality metric. Manually evaluating all clusters found in our experiments in order to check the coherence of the scores would be prohibitively expensive. So, we compare the results obtained using our metric with results from two classic quality metrics: *modularity*, which is one of the most popular metrics in the literature, and *silhouette*, which uses different graph characteristics to measure cluster quality.

Also, to better understand how our proposed metric reacts when applied to different kinds of complex networks, we will show some results obtained for social, technological and biological networks. For simplicity, only a subset from the datasets studied will be discussed here, but the behaviors we describe, unless otherwise noted, are similar for all networks of the same type.

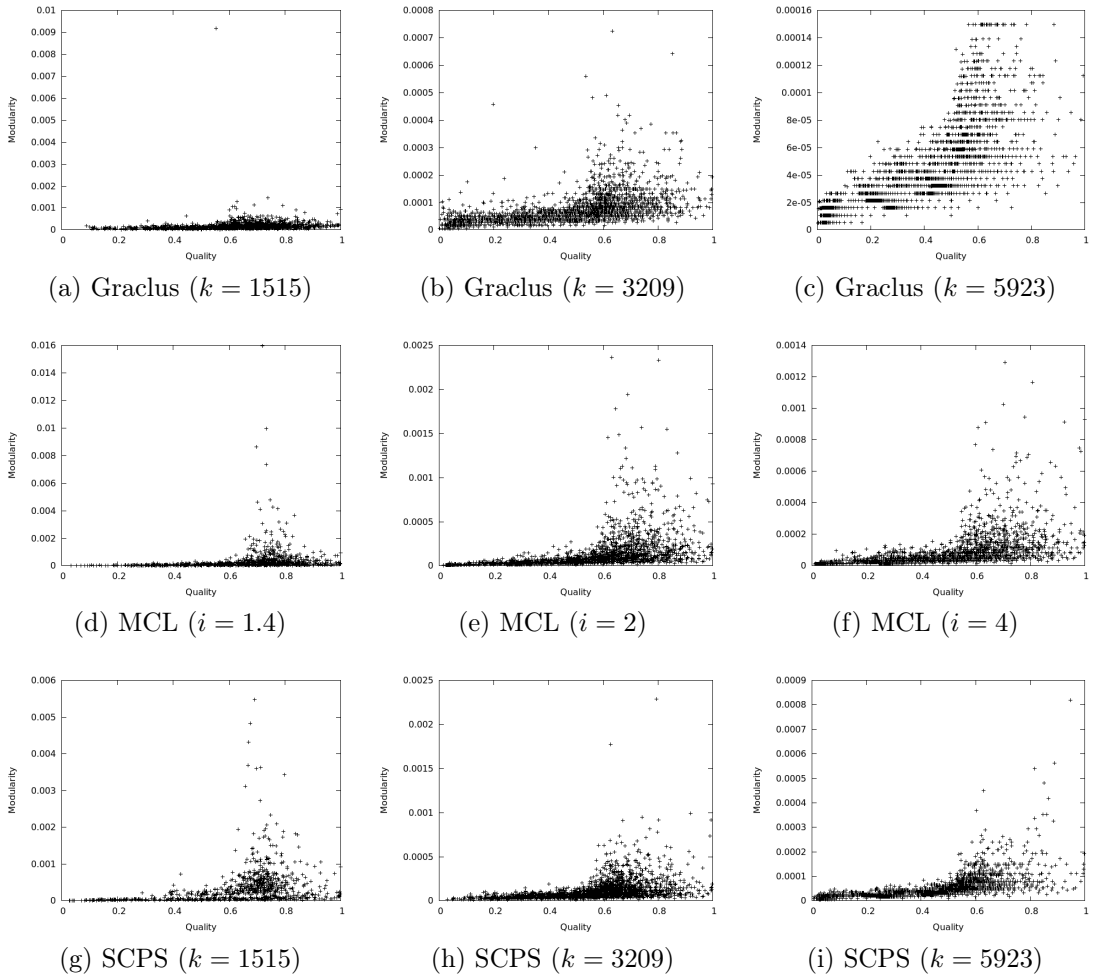
6.2.1 Performance for Social Networks

We first discuss how our metric performs when applied to graphs from social networks, using results obtained from the Condensed Matter collaboration network. Figure 6.1 presents a comparison between our proposed metric and modularity through a series of scatter plots, where each graph represents the results of a clustering produced by a given algorithm, using a given set of parameters. In each graph, each point represents one cluster found, and its position shows it's score for both our metric (X axis) and modularity (Y axis). One interesting thing that can be seen in that figure is that there is a significant number of clusters that contribute little to the overall modularity score (i.e., have low modularity), but that our metric still considers highly. On the other hand, there are just a few clusters that behave the opposite way.

	# Vert.	**# Edg.**	**Mod**	**SI**	**Cond.**	**IDI**	**P. Metric**
# Vert.	1	0.48	0.47	0.039	-0.17	0.071	-0.075
# Edg.	0.48	1	0.99	-0.023	-0.16	0.59	0.3
Mod	0.47	0.99	1	-0.024	-0.15	0.59	0.31
SI	0.039	-0.023	-0.024	1	0.11	-0.048	0.02
Cond.	-0.17	-0.16	-0.15	0.11	1	-0.036	0.42
IDI	0.071	0.59	0.59	-0.048	-0.036	1	0.55
P. Metric	-0.075	0.3	0.31	0.02	0.42	0.55	1

Table 6.2: Kendall's Tau for the Condensed Matter Collaboration network results, clustered by Graclus with $k = 1515$.

To better understand those cases, we can evaluate the ranking behavior of those metrics using Kendall's Tau correlation index. Table 6.2 presents the correlation indexes not only between the quality metrics considered, but also cluster size (number of vertices), internal edge count, and the internal density index (IDI), the internal quality component of our metric. We chose the results for Graclus with $k = 1515$ for it presents easily identifiable clusters with differing scores for the two metrics evaluated.



(a) Graclus ($k = 1515$) (b) Graclus ($k = 3209$) (c) Graclus ($k = 5923$)

(d) MCL ($i = 1.4$) (e) MCL ($i = 2$) (f) MCL ($i = 4$)

(g) SCPS ($k = 1515$) (h) SCPS ($k = 3209$) (i) SCPS ($k = 5923$)

Figure 6.1: Our metric versus Modularity for clusters from the Condensed Matter Collaboration dataset. Ranges on the Y axis vary.

From Table 6.2, we can see that modularity and our metric present just a small level of agreement on their rankings, a behavior that can also be seen in Figure 6.1a, where a great number of clusters with low modularity have high scores of our metric. Other interesting information that can be gathered is that modularity ranks almost the same as just sorting the clusters by their number of internal edges. This comes as no surprise, as our previous studies showed that modularity considered internal density to

be just the number of internal edges. Also, silhouette shows a ranking behavior that is almost independent from the rest of the studied metrics.

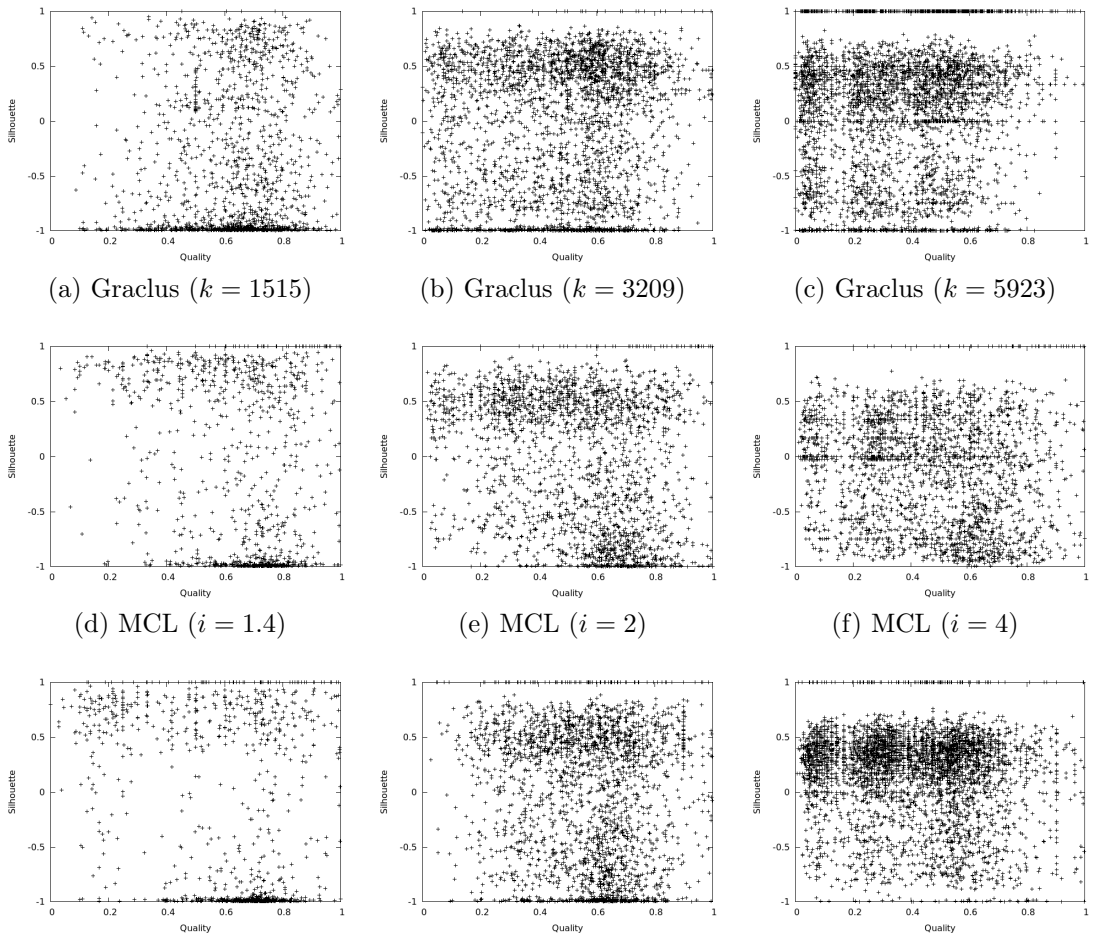
C. ID	# Vert.	# Edg.	Mod	SI	Cond.	IDI	P. Metric
457	977	2326	0.0092	-0.99	0.88	0.99	0.55
386	29	274	0.0015	0.68	0.52	1	0.74
584	41	240	0.0013	-0.017	0.74	1	0.63
714	31	212	0.0011	-0.99	0.62	1	0.69
690	28	208	0.0011	-0.49	0.42	1	0.79
226	27	200	0.0011	-0.99	0.76	1	0.62
682	40	198	0.0011	-0.98	0.78	1	0.61
617	32	179	0.00095	-0.96	0.75	1	0.63
1126	21	174	0.00093	0.29	0	1	1
283	26	171	0.00091	-0.99	0.32	1	0.84

Table 6.3: Best 10 clusters (modularity-wise) from the Condensed Matter Collaboration dataset, clustered by Graclus with $k = 1515$.

Even though the correlation index does confirm our suspicion about the high level of disagreement between the scoring done by our metric and modularity, it does not shed any light on which of the two metrics is correct on contested cases, such as the lone point at the top of Figure 6.1a, which has the best modularity score for Graclus with $k = 1515$, but is considered barely good by our metric, for example.

To answer this question, Table 6.3 presents some information about the 10 clusters with the highest modularity scores for that particular experimented configuration. It is possible to see that the cluster previously discussed, with ID 457, is indeed internally dense. With 977 vertices and 2326 edges, even our internal quality index agrees that this is among the densest clusters of this size that may exist in that network. However, conductance shows us that that cluster is extremely connected to the rest of the graph, with 88% of its total edges being external. So, even though fairly internally dense, this cluster cannot be considered well formed. Overall, modularity considers it to be the best in this clustering, silhouette and conductance give it bad scores and our metric says that it is borderline good.

Scores from our metric and silhouette rarely agree on the quality of a given cluster, with their correlation index, shown in Table 6.2, indicating that their scoring behavior is almost independent. We can see good examples of their disagreement on clusters 283 and 690 from Table 6.3, which are both fairly dense and externally sparse, receiving good scores from our metric, while at the same time receiving very low silhouette scores. Based on the results from Figure 6.2, which presents the comparison between the scores from our metric and silhouette for different clustering scenarios, it is possible



(a) Graclus ($k = 1515$) (b) Graclus ($k = 3209$) (c) Graclus ($k = 5923$)

(d) MCL ($i = 1.4$) (e) MCL ($i = 2$) (f) MCL ($i = 4$)

(g) SCPS ($k = 1515$) (h) SCPS ($k = 3209$) (i) SCPS ($k = 5923$)

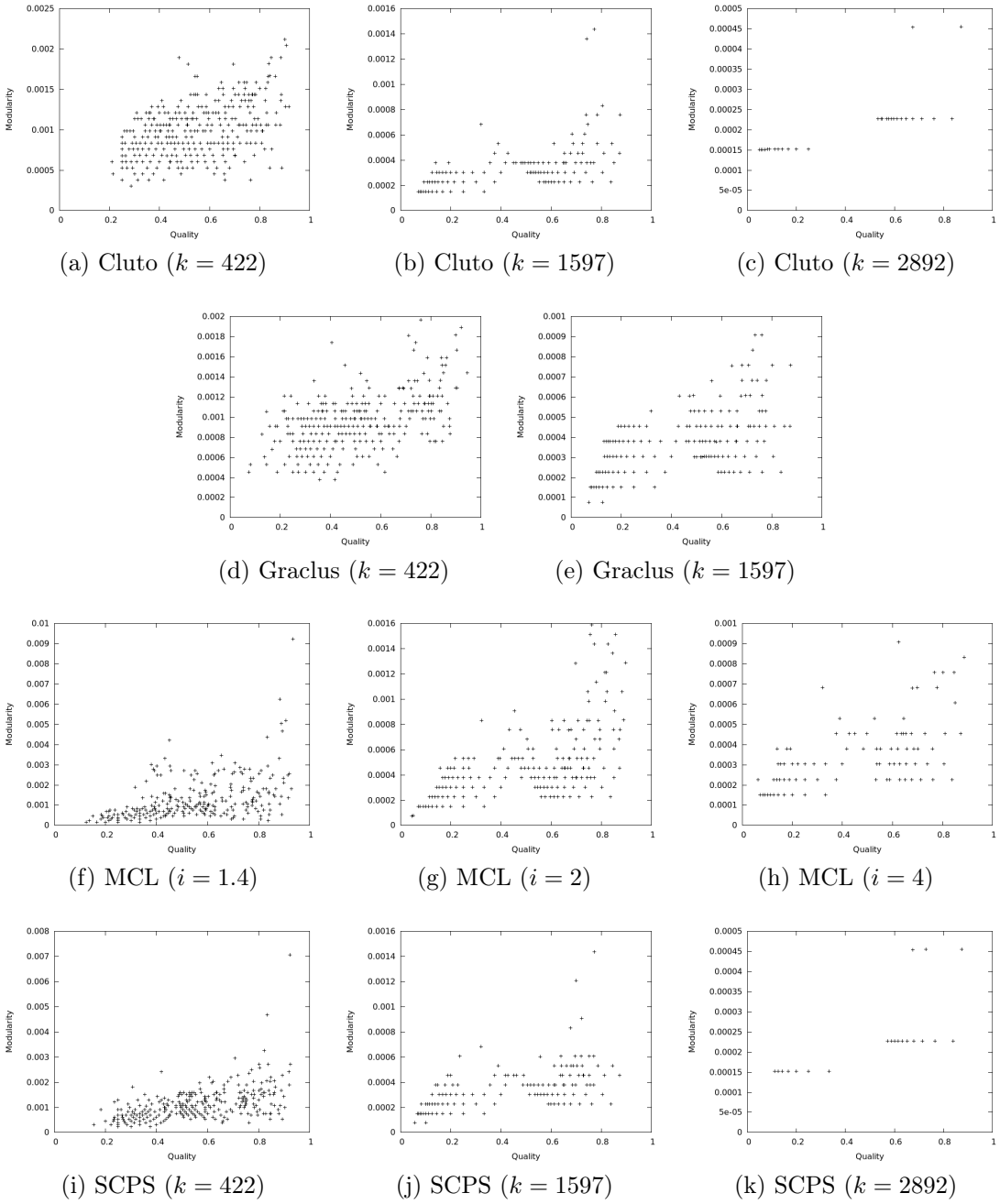
Figure 6.2: Our proposed metric versus Silhouette for clusters from the Condensed Matter Collaboration network dataset.

to see that those two metrics present no discernible correlation between their scoring patterns.

6.2.2 Performance for Technological Networks

To evaluate the behavior of our metric when applied to technological networks, we use results obtained from the Power Grid network. In Figure 6.3, we compare the results of our metric and modularity for that network. Again, scores for our metric are on the X axis and for modularity on the Y axis. It is important to notice that the results for the finest cluster granularities evaluated (Figures **??**, 6.3e, 6.3h and 6.3k) do not always show results for all clusters found on those test scenarios. This happens because most of the clusters found are too small, with the case presented on Figure 6.3k, for

example, having only 198 out of its 2892 clusters with 3 or more vertices. Clusters with only 1 or 2 vertices present problems to the sampling process of IDI, as they normally lack a large enough universe of possible clusters to be sampled. So, those cases become impossible to evaluate through our metric. Nonetheless, since they are too small to be considered good clusters anyway, this poses no great problem for our evaluation.



(a) Cluto ($k = 422$) (b) Cluto ($k = 1597$) (c) Cluto ($k = 2892$)

(d) Graclus ($k = 422$) (e) Graclus ($k = 1597$)

(f) MCL ($i = 1.4$) (g) MCL ($i = 2$) (h) MCL ($i = 4$)

(i) SCPS ($k = 422$) (j) SCPS ($k = 1597$) (k) SCPS ($k = 2892$)

Figure 6.3: Our proposed metric versus Modularity for clusters from the Power Grid dataset. Ranges on the Y axis are variable.

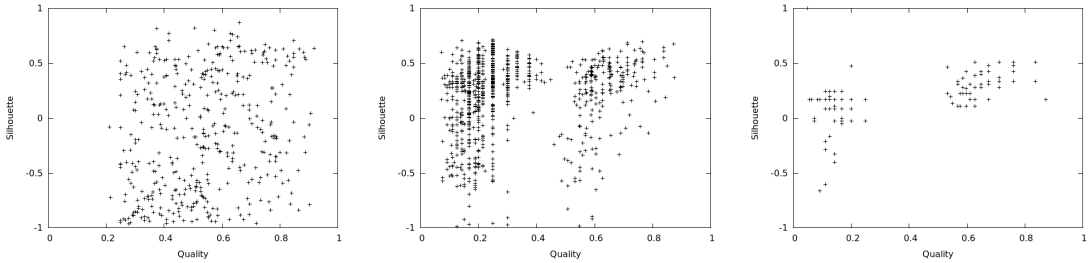
C. ID	# Vert.	# Edg.	Mod	SI	Cond.	IDI	P. Metric
1	63	123	0.0092	-0.76	0.1	0.97	0.94
2	54	83	0.0062	-0.9	0.19	0.96	0.88
4	49	69	0.0052	0.099	0.1	0.92	0.91
39	26	67	0.005	-0.84	0.22	1	0.89
5	45	62	0.0047	-0.97	0.1	0.89	0.89
6	42	58	0.0044	-0.95	0.24	0.91	0.83
3	53	56	0.0042	-0.35	0.19	0.088	0.45
8	39	46	0.0035	-0.92	0.18	0.49	0.65
36	27	44	0.0033	-0.85	0.21	0.98	0.88
7	41	44	0.0033	-0.89	0.33	0.2	0.43

Table 6.4: Best 10 clusters (modularity-wise) of the Power Grid dataset, clustered by MCL with $i = 1.4$.

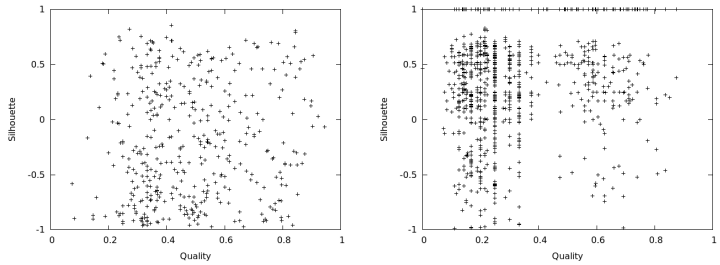
One interesting thing to notice is that there is a slightly bigger agreement between what those two quality metrics believe to be the best clusters, at least for the coarsest settings tested (Figures 6.3a, 6.3d, 6.3f and 6.3i). Even so, there are still many clusters where the disagreement between those two metrics is strong. On Table 6.4, we present the 10 clusters with highest modularity found by MCL with $i = 1.4$, a case that clearly presented this kind of disagreement. It is possible to observe that the top 6 clusters, modularity-wise, are also very well regarded by our metric. However, it can also be seen that modularity's bias in favor of clusters with high internal edge counts still makes it score positively clusters that are barely denser than trees (IDs 3 and 7, for example). On the other hand, our metric correctly penalizes those same clusters because of their internal sparsity.

Concerning the silhouette index, once again it and our metric present a great level of disagreement, with relatively few clusters having consistent scores for the two metrics. To better understand the disagreements between their scores, Table 6.5 presents the clusters with higher silhouette found by MCL with $i = 1.4$, the same scenario evaluated for modularity, and one which presents clusters with high silhouette and low scores of our quality metric. It is possible to see that silhouette tends to favor very small clusters, even though they are almost tree-like, given their internal edge counts.

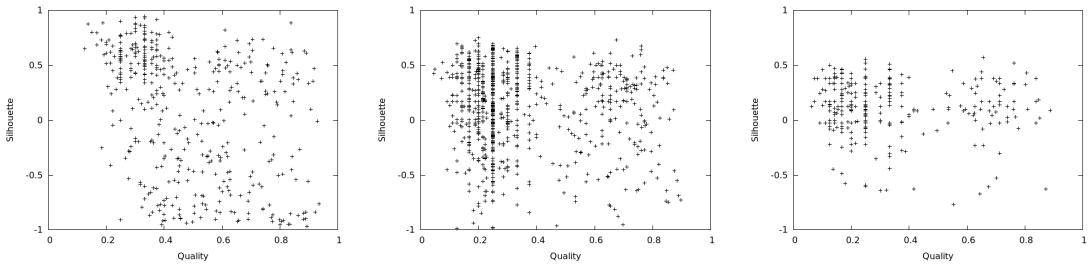
One interesting point is the situation of cluster 402 from that same table, which presents a very high score for our metric, even though it is very small. This cluster is a 3-clique connected to the rest of the graph by only one edge, as can be seen by its conductance score. Considering the canonical structure of a cluster, this is the best configuration a cluster with 3 vertices can have without being disconnected from the rest of the graph. However, this does not mean that a cluster this small is interesting.



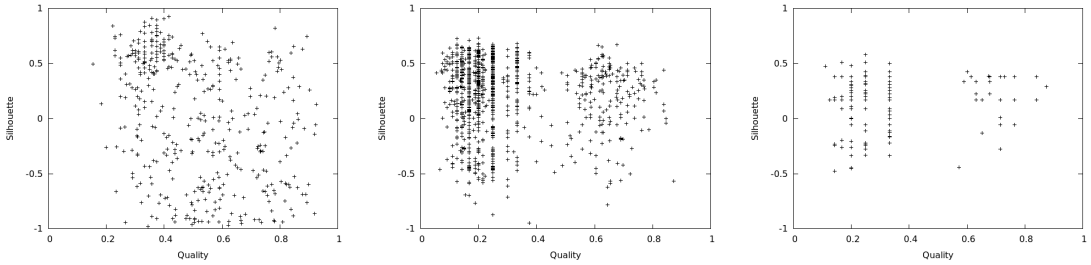
(a) cluto-k422 (b) cluto-k1597 (c) cluto-k2892



(d) graclus.part.422 (e) graclus.part.1597



(f) MCL-i1.4 (g) MCL-i2 (h) MCL-i4



(i) scps-k422 (j) scps-k1597 (k) scps-k2892

Figure 6.4: Our proposed metric versus Silhouette for clusters from the Power Grid dataset.

Some ways to avoid giving good scores for structures like that could be excluding clusters with too few vertices, or to give a scoring advantage for larger clusters, for example.

Clust. ID	# Vert.	# Edg.	Mod	SI	Cond.	IDI	P. Metric
403	3	2	0.00015	0.94	0.33	0	0.33
369	4	3	0.00023	0.94	0.4	0	0.3
396	3	2	0.00015	0.93	0.33	0	0.33
387	4	3	0.00023	0.92	0.25	0	0.38
355	5	4	0.0003	0.9	0.5	0	0.25
402	3	3	0.00023	0.89	0.25	0.93	0.84
247	7	6	0.00045	0.88	0.62	0	0.19
377	4	3	0.00023	0.88	0.4	0	0.3
371	4	3	0.00023	0.88	0.73	0	0.14
386	4	3	0.00023	0.87	0.4	0	0.3

Table 6.5: Best 10 clusters (silhouette-wise) of the Power Grid dataset, clustered by MCL with $i = 1.4$.

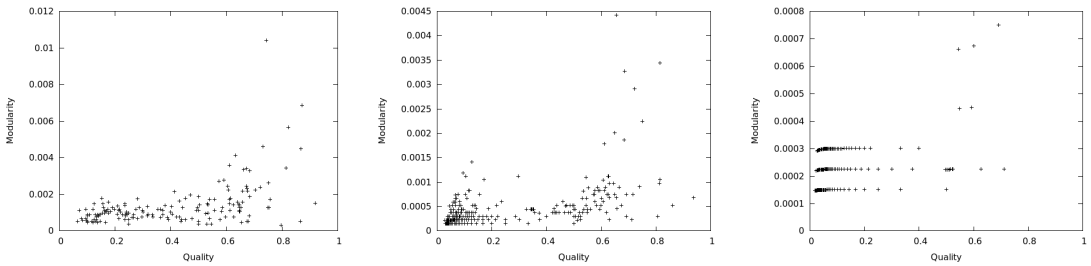
6.2.3 Performance for Biological Networks

To evaluate our metric's performance when applied to biological networks, we use the results from the Yeast protein interaction dataset. Figure 6.5 presents some of those results, with scores from our metric on the X axis and modularity on the Y axis as before. It is possible to see that the disagreement which occurred in the other kinds of networks still persists, with clusters of low modularity receiving relatively high scores of our metric.

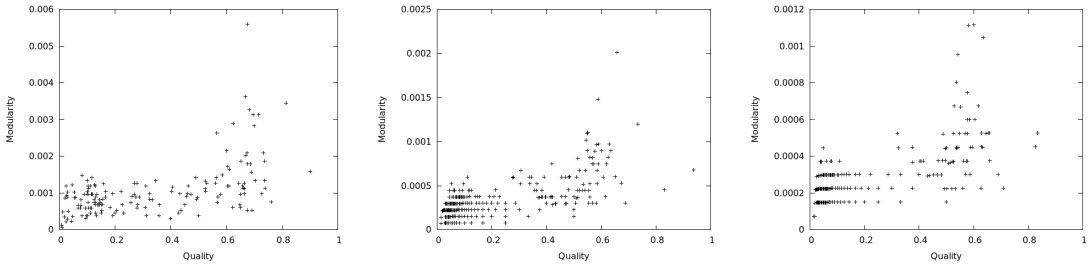
Clust. ID	# Vert.	# Edg.	Mod	SI	Cond.	IDI	P. Metric
69	12	20	0.0015	-0.65	0.13	0.97	0.92
14	34	91	0.0068	-0.93	0.25	1	0.87
87	6	7	0.00053	0.74	0.12	0.85	0.86
13	34	64	0.0048	0.027	0.27	0.99	0.86
23	29	106	0.0079	-0.76	0.38	1	0.81
11	37	73	0.0054	-0.85	0.39	0.99	0.8
77	10	15	0.0011	-0.55	0.35	0.94	0.79
25	28	68	0.0051	-0.93	0.45	1	0.78
10	38	117	0.0086	-0.53	0.47	1	0.76
1	109	387	0.027	-0.94	0.51	1	0.74

Table 6.6: Best 10 clusters (quality-wise) of the Yeast dataset, clustered by SCPS with $k = 167$.

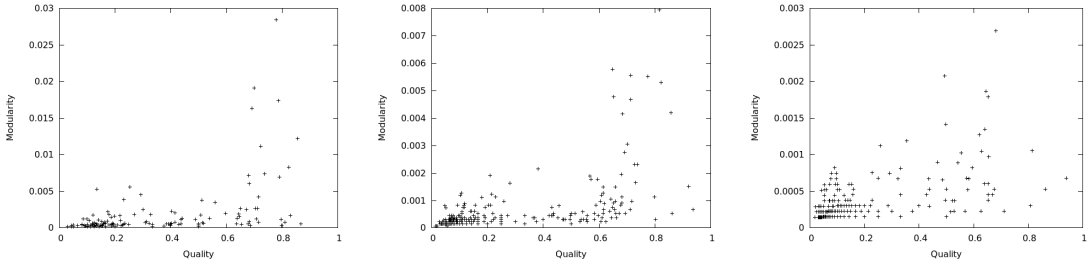
Table 6.6 presents the 10 clusters with better scores by our metric. Through those results, it is possible to see that, in general, the best cluster for our metric are also the ones with highest modularity. However, in some cases, those two metrics disagree. Cluster 87 presents a high score of our metric and silhouette, although it



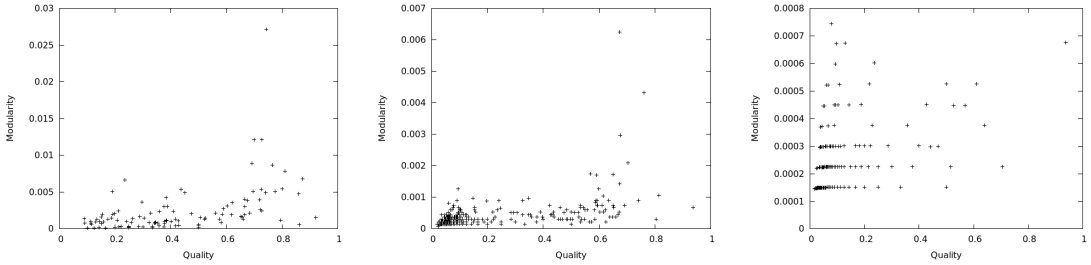
(a) Cluto ($k = 167$) (b) Cluto ($k = 521$) (c) Cluto ($k = 910$)



(d) Graclus ($k = 167$) (e) Graclus ($k = 521$) (f) Graclus ($k = 910$)



(g) MCL ($i = 1.4$) (h) MCL ($i = 2$) (i) MCL ($i = 4$)



(j) SCPS ($k = 167$) (k) SCPS ($k = 521$) (l) SCPS ($k = 910$)

Figure 6.5: Our proposed metric versus Modularity for clusters from the Yeast dataset. Ranges on the Y axis are variable.

has low modularity. The low modularity is easily explained by the cluster's small edge count. But, with only 6 vertices and 7 edges, cluster 87 seems like a bad fit for a well structured cluster.

Looking at the structure of cluster 87, which can be seen in Figure 6.6, it is possible to notice that all its vertices are connected to a "hub", which also connects

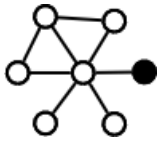


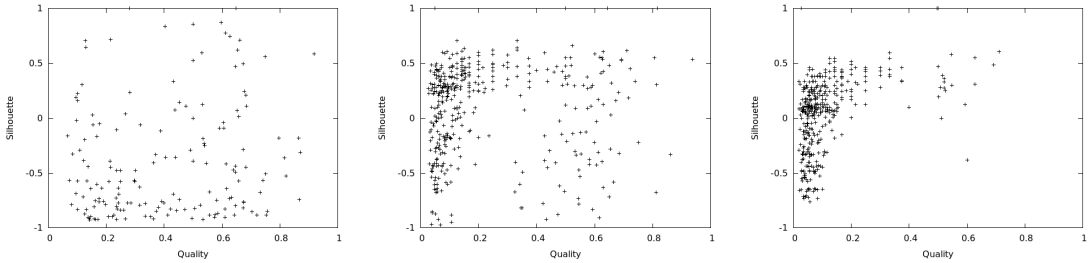
Figure 6.6: Structure of cluster 87 from SCPS with $k = 167$ (Yeast dataset). White vertices belong to the cluster, and the black vertex represents the rest of the graph.

the cluster to the rest of the graph. Silhouette scores it positively because of the short paths between member vertices caused by the hub, and our metric does so because it is denser than most 6-vertex subgraphs in this graph, while also being very sparsely connected to the rest of the graph. This kind of structure is more similar to the type of cluster described in [Zaidi et al., 2010], and, as the authors of that paper state, are better evaluated by metrics that use vertex distance as the similarity function, like silhouette does. Our metric considers it to be good not by evaluating it as an structure similar to the one Zaidi et al. defend to be good, but because it fits the canonical view as much as this specific graph can, with its sparser nature. Had that cluster fewer internal edges, our metric would probably consider it inadequate, even if silhouette kept scoring it positively.

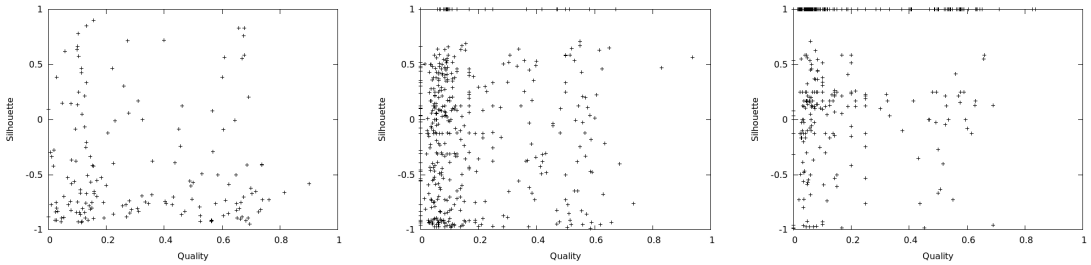
The comparison between our metric and silhouette can be seen on Figure 6.7. In this case, the same trends observed in the other kinds of networks are visible. There is a great level of disagreement between those two metrics, and most of the clusters where they agree are considered bad by both of them. For the finer clusterings (shown in the third column of that figure), silhouette keeps the behavior discussed on Chapter 4, with the tendency to give better scores to smaller clusters, which are much more common at that granularity. Our metric, on the other hand, has a greater tendency to give worse scores for those cases. When taking a closer look at the best silhouette results for the finest granularity setting of the SCPS experiments (Figure 6.7l), a test configuration with easy to find clusters with perfect silhouette and low scores of our metric, it is possible to see that those excellent silhouette clusters had only one vertex each, falling on silhouette's "blind spot".

6.2.4 Overview of Our Proposed Metric's Scoring Behavior

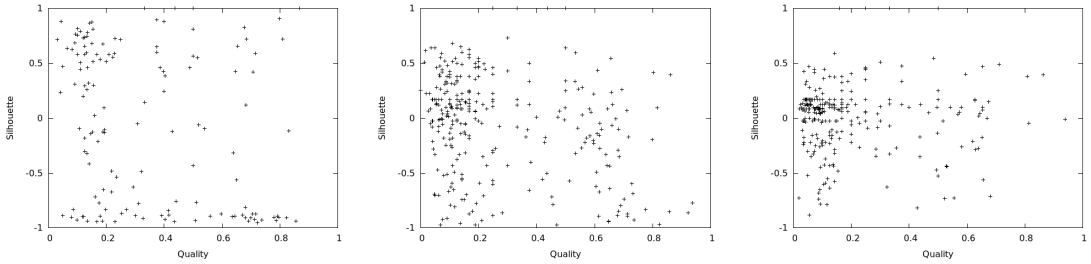
We now discuss the scoring behavior of our proposed metric. As we saw in Chapter 4, the clustering quality metrics commonly used in the literature present some bias towards clusters of a given size, even to the detriment of the kind of structure classically associated with good clusters. We want to be sure that our metric does not have such



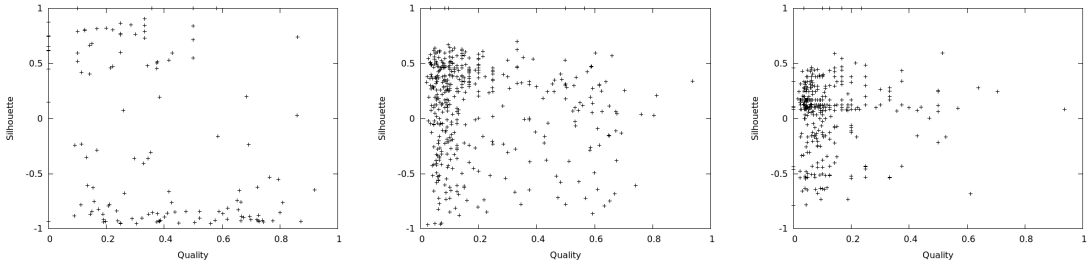
(a) Cluto ($k = 167$) (b) Cluto ($k = 521$) (c) Cluto ($k = 910$)



(d) Graclus ($k = 167$) (e) Graclus ($k = 521$) (f) Graclus ($k = 910$)



(g) MCL ($i = 1.4$) (h) MCL ($i = 2$) (i) MCL ($i = 4$)

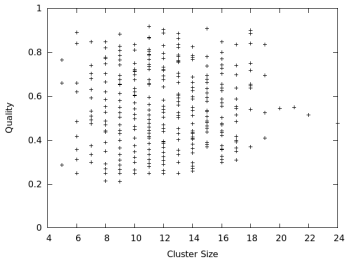


(j) SCPS ($k = 167$) (k) SCPS ($k = 521$) (l) SCPS ($k = 910$)

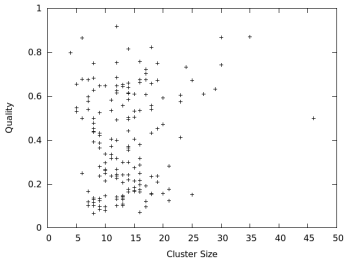
Figure 6.7: Our proposed metric versus Silhouette for clusters from the Yeast dataset.

bias, assigning its scores fairly.

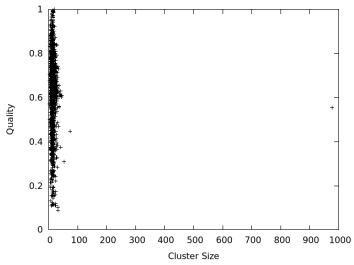
To do so, we study how our metric evaluates clusters given their size. Figure 6.8 presents the results for our metric by cluster size, for all clustering algorithms evaluated on their coarsest settings, applied to the Condensed Matter Collaboration (social), Power Grid (technological) and Yeast (biological) networks. We chose those settings because they presented the widest range of cluster sizes for each clustering. Based on



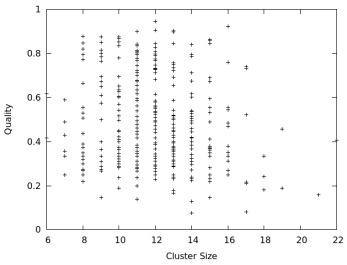
(a) Power Grid (Cluto $k = 422$)



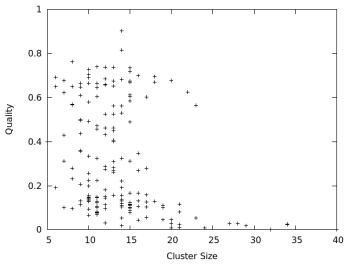
(b) Yeast (Cluto $k = 167$)



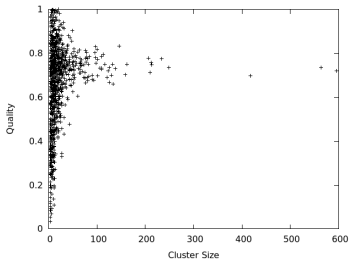
(c) Cond. Mat. (Graclus $k = 1515$)



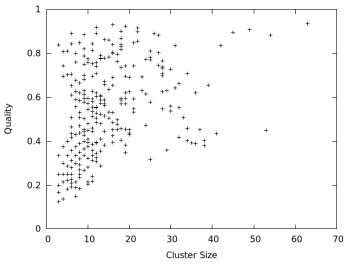
(d) Power Grid (Graclus $k = 422$)



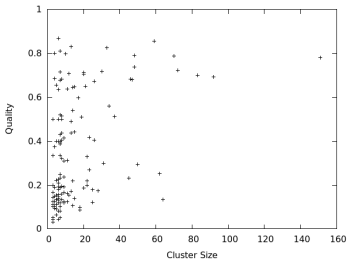
(e) Yeast (Graclus $k = 167$)



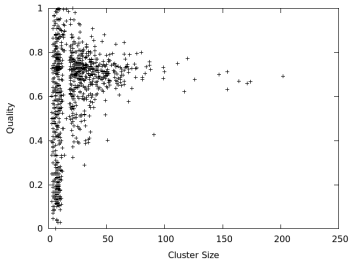
(f) Cond. Mat. (MCL $i = 1.4$)



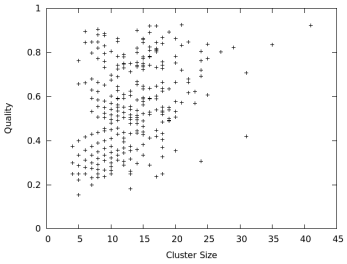
(g) Power Grid (MCL $i = 1.4$)



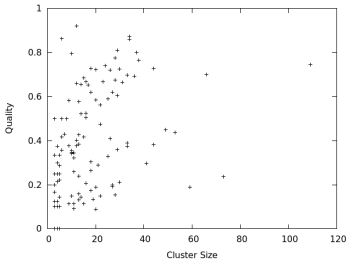
(h) Yeast (MCL $i = 1.4$)



(i) Cond. Mat. (SCPS $k = 1515$)



(j) Power Grid (SCPS $k = 422$)

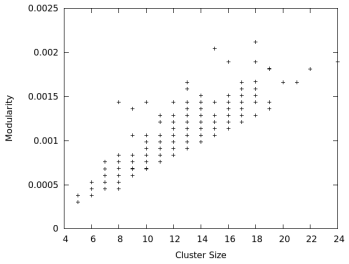


(k) Yeast (SCPS $k = 167$)

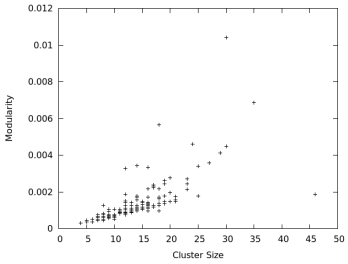
Figure 6.8: Quality by cluster size for all clustering algorithms, using their coarsest settings, applied to the Condensed Matter Collaboration, Power Grid and Yeast datasets.

those results, it is possible to see that, no matter the size or origin of the network, a large part of the clusters found is rather small, with only a few clusters with more than 50 vertices. Also, it is noticeable that the cluster size has a small impact on quality

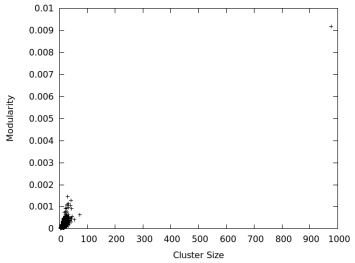
scores, specially when compared to the results obtained from modularity (Figure 6.9), which show a very consistent tendency to give better scores for larger clusters, no matter what.



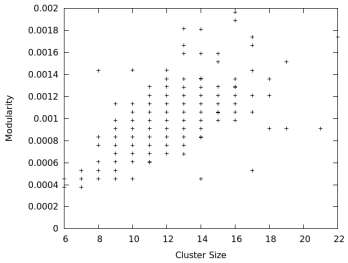
(a) Power Grid (Cluto $k = 422$)



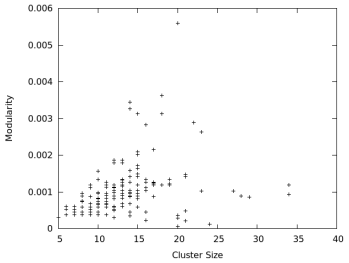
(b) Yeast (Cluto $k = 167$)



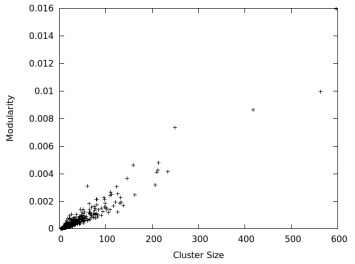
(c) Cond. Mat. (Graclus $k = 1515$)



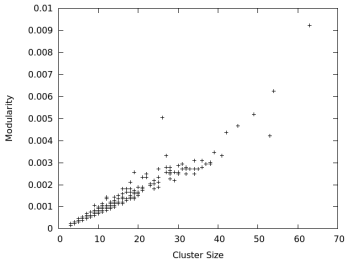
(d) Power Grid (Graclus $k = 422$)



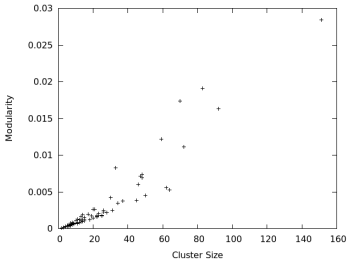
(e) Yeast (Graclus $k = 167$)



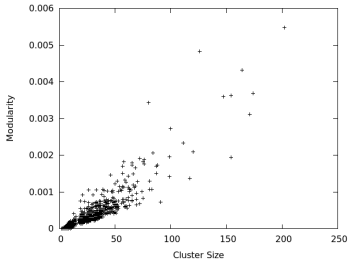
(f) Cond. Mat. (MCL $i = 1.4$)



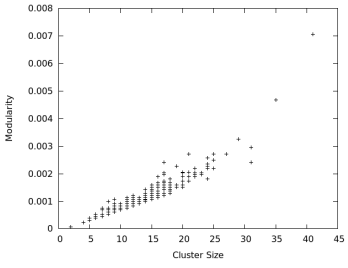
(g) Power Grid (MCL $i = 1.4$)



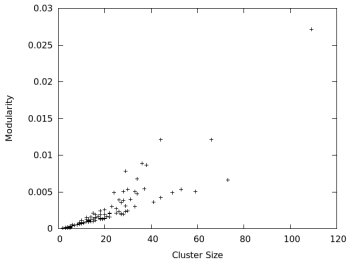
(h) Yeast (MCL $i = 1.4$)



(i) Cond. Mat. (SCPS $k = 1515$)



(j) Power Grid (SCPS $k = 422$)



(k) Yeast (SCPS $k = 167$)

Figure 6.9: Modularity by cluster size for all clustering algorithms, using their coarsest settings, applied to the Condensed Matter Collaboration, Power Grid and Yeast datasets.

This tendency of modularity is expected, as larger clusters have a greater probability of also having more edges to connect their vertices, and modularity is very biased towards larger internal edge counts. Our metric showed no such biases, the exception being found on the clusters from the social network, where it has the tendency to also give better scores for the larger clusters. However, since this behavior is consistent for all clustering methods, at the same time that it does not occur as strongly on any other studied networks, social or not, it is probably a particular tendency of that network itself, and not from our metric.

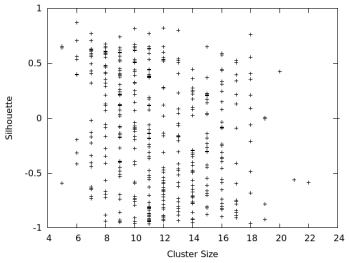
As for the silhouette index, its results show a slight bias towards giving worse results to larger clusters, and better results to small ones. However, this bias is nowhere as strong as the one displayed by modularity. It is important to notice that, as clusterings get larger (i.e., with more clusters), each cluster will get smaller, and the biases shown here will only increase in strength, as described in Chapter 4.

6.2.5 Evaluation of Internal and External Quality

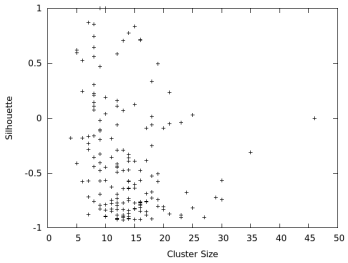
By isolating the internal and external components of our metric, it is possible to compare the different clustering algorithms used in our experiments. This kind of evaluation is helpful not only to understand the differences between clustering algorithms, but also to better comprehend the inner workings of our metric. We will first discuss our quality results for the clusterings of the Condensed Matter Collaboration dataset (a social network). Figure 6.11 presents scatter plots of the solutions obtained by the clustering algorithms mentioned before, with each dot representing a cluster's internal (X axis) and external (Y axis) quality values. In those plots, the best scored clusters are those found in the lower right quadrant (higher internal and lower external densities).

One important thing that can be seen from Figure 6.11's plots is that Graclus, MCL and SCPS present a rather similar clustering behavior. Most of the clusters found by those algorithms are not good in at least one of the two cluster characteristics. This is expected, as it is unreasonable to believe that all elements of the network are part of one, and only one, well structured cluster. The graphs studied are snapshots of very dynamic networks and, because of that, will always be incomplete. However, this is one more reminder that evaluating only one of the two main graph characteristics will undoubtedly generate untrustworthy results.

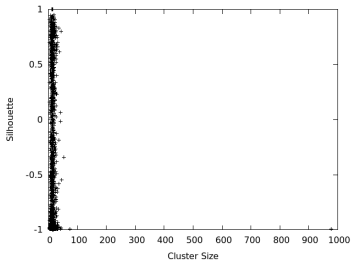
Another interesting insight is that, for those three clustering algorithms, most of the clusters found were among the densest possible for this graph, as can be seen by their internal quality indexes. Those clusters, however, frequently are not well separated from the rest of the graph, as can be seen from their external quality in-



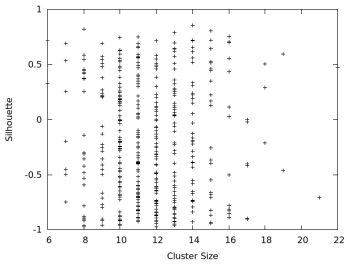
(a) Power Grid (Cluto $k = 422$)



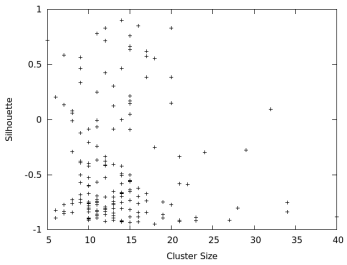
(b) Yeast (Cluto $k = 167$)



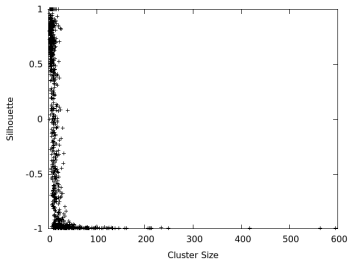
(c) Cond. Mat. (Graclus $k = 1515$)



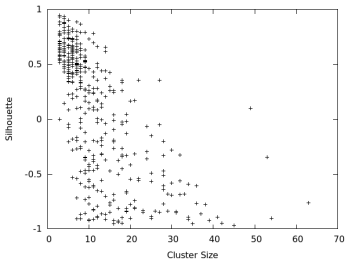
(d) Power Grid (Graclus $k = 422$)



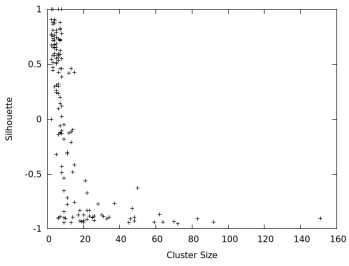
(e) Yeast (Graclus $k = 167$)



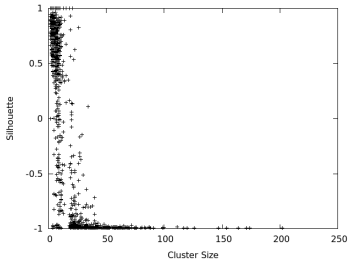
(f) Cond. Mat. (MCL $i = 1.4$)



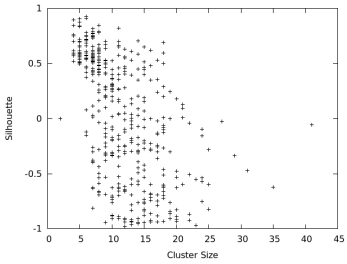
(g) Power Grid (MCL $i = 1.4$)



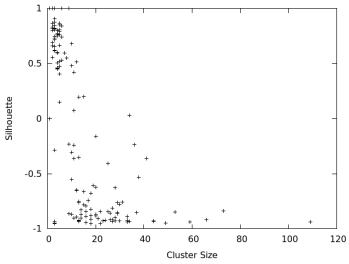
(h) Yeast (MCL $i = 1.4$)



(i) Cond. Mat. (SCPS $k = 1515$)



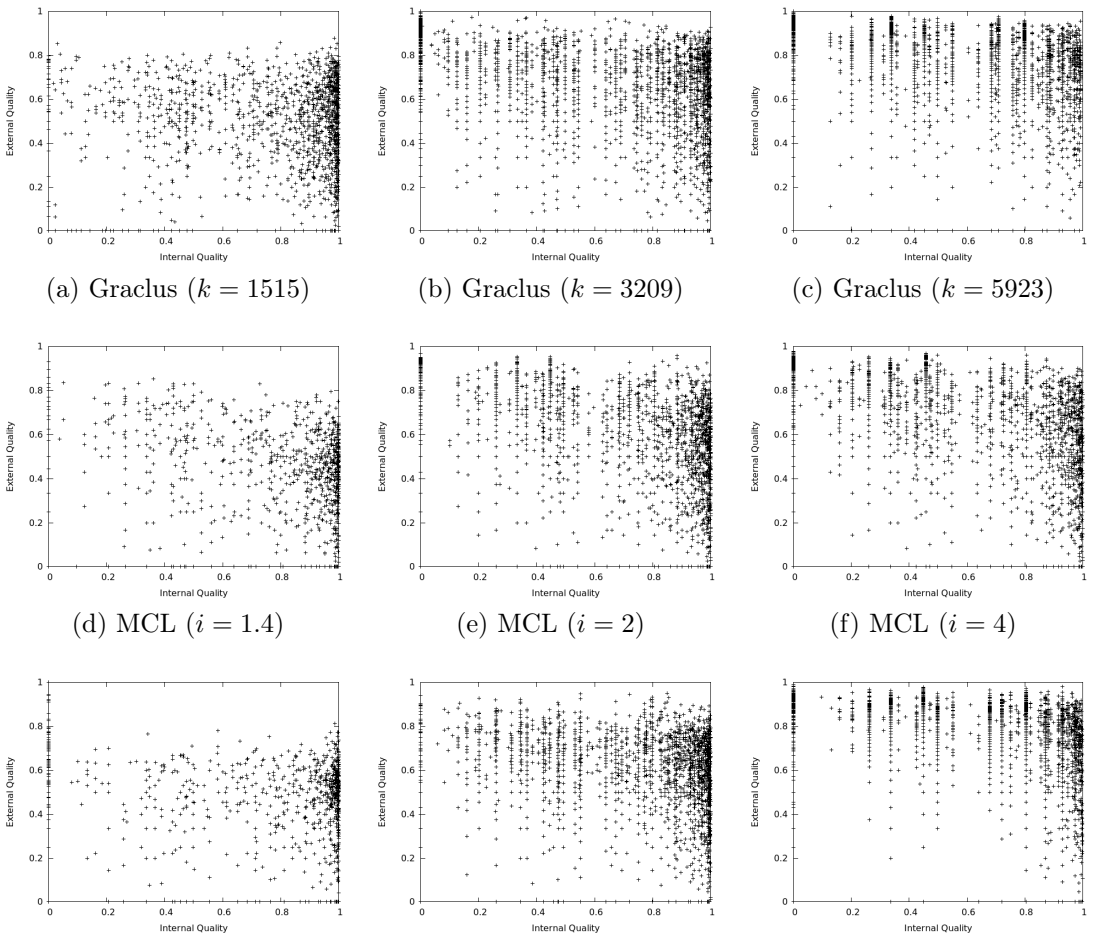
(j) Power Grid (SCPS $k = 422$)



(k) Yeast (SCPS $k = 167$)

Figure 6.10: Silhouette by cluster size for all clustering algorithms, using their coarsest settings, applied to the Condensed Matter Collaboration, Power Grid and Yeast datasets.

dexes (conductance), and this lack of well defined separation gets more acute for finer clusterings (i.e., with smaller clusters). Considering that conductance is given by the



(a) Graclus ($k = 1515$) (b) Graclus ($k = 3209$) (c) Graclus ($k = 5923$)

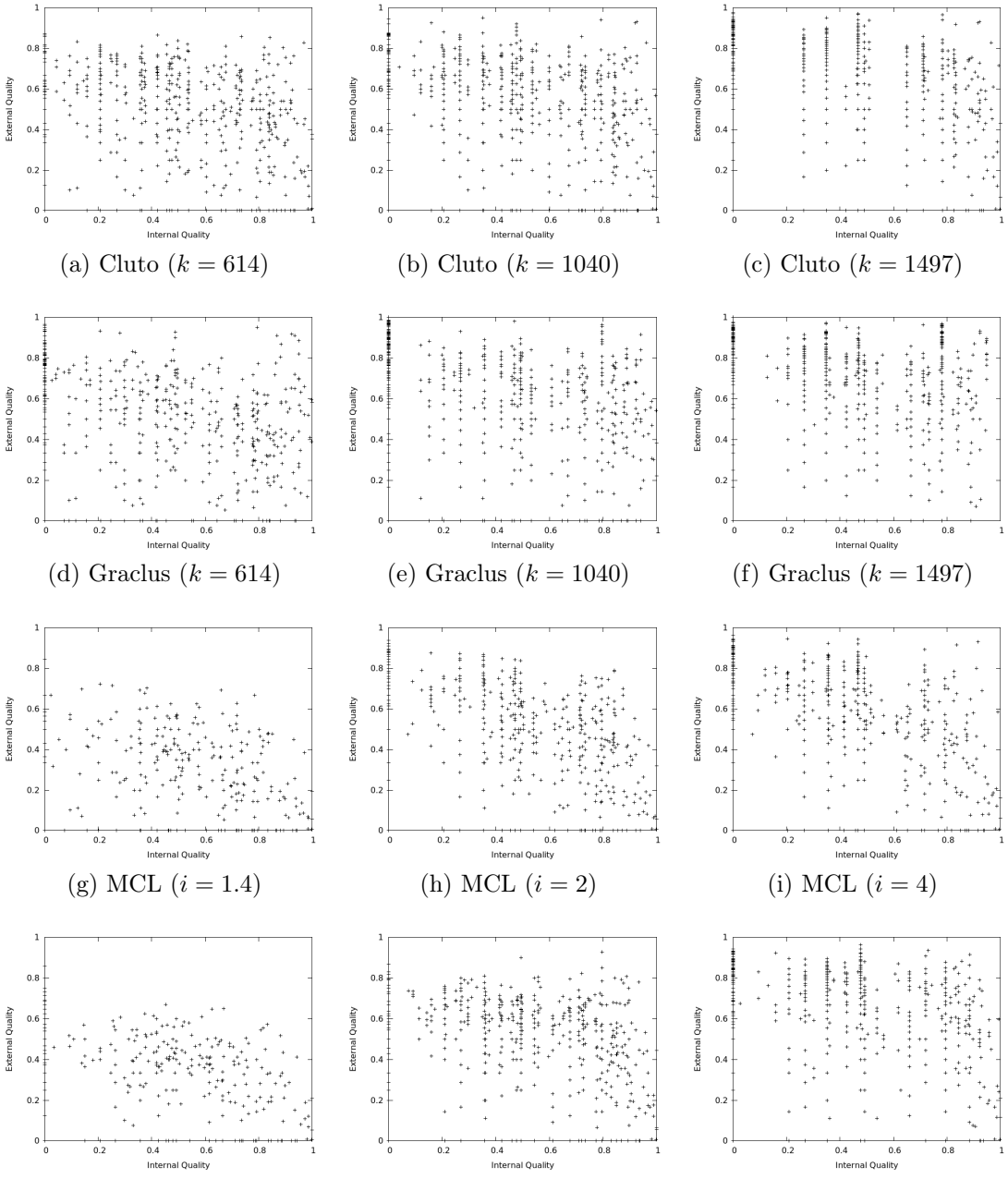
(d) MCL ($i = 1.4$) (e) MCL ($i = 2$) (f) MCL ($i = 4$)

(g) SCPS ($k = 1515$) (h) SCPS ($k = 3209$) (i) SCPS ($k = 5923$)

Figure 6.11: Internal vs. external quality values for clusters from the Condensed Matter Collaboration dataset.

proportion of all edges with at least one endpoint inside the evaluated cluster that connect it to vertices outside, it is fair to assume that it is easier for smaller clusters to have worse conductance values, as they need fewer external edges to cause conductance values to rise. This behavior is shared by almost all of the social networks evaluated, with the exception being General Relativity Collaboration, as can be seen on Figure 6.12. For this particular network, the four clustering algorithms used presented no discernible tendency on the characteristics of clusters found. So, even when working with networks of similar origins, each one of them might present unique characteristics that are hard to predict. One more reason to use the network itself in order to establish what kind of cluster structure is interesting or not.

Also, as the granularity of the clustering gets finer, the resulting clusters have a tendency to be either very internally dense or very internally sparse, with very few



(a) Cluto ($k = 614$) (b) Cluto ($k = 1040$) (c) Cluto ($k = 1497$)

(d) Graclus ($k = 614$) (e) Graclus ($k = 1040$) (f) Graclus ($k = 1497$)

(g) MCL ($i = 1.4$) (h) MCL ($i = 2$) (i) MCL ($i = 4$)

(j) SCPS ($k = 614$) (k) SCPS ($k = 1040$) (l) SCPS ($k = 1497$)

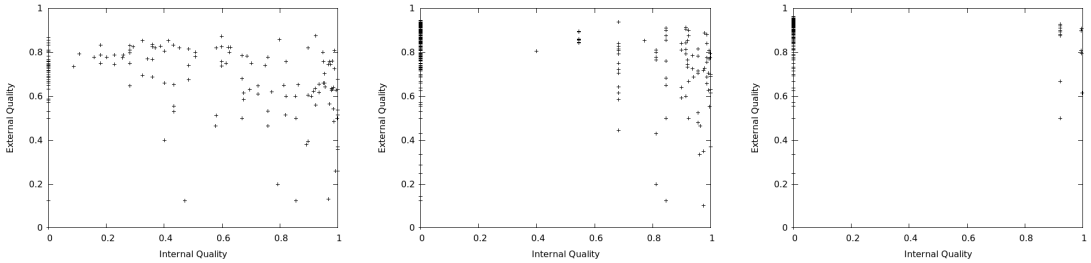
Figure 6.12: Internal vs. external quality values for clusters from the General Relativity Collaboration network dataset.

clusters staying in-between those two extremes. Their external sparsity levels, however, still present a reasonable level of variation, regardless of clustering granularity. Those two behaviors were observed for all networks studied, with the two collaboration networks showing them in a weaker form than the rest. The behavior of internal densities raises an interesting point: are clusters found in real networks formed by very dense

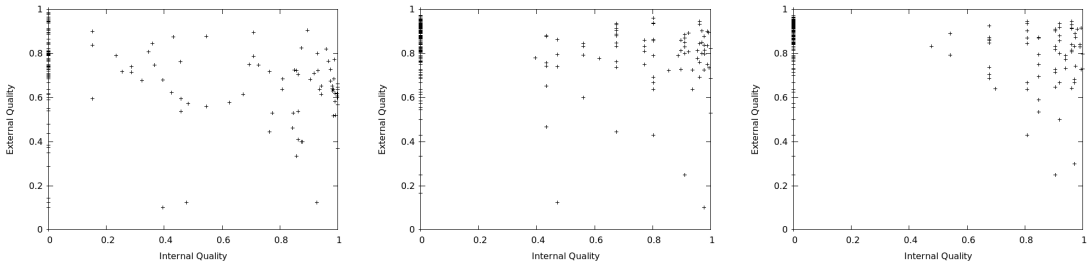
cores, which get gradually sparser as vertices get farther away from them? If that is the case, then finer clusterings would probably cause the removal of the outer "layers" of the core clusters, keeping the cores relatively intact and generating new, very small clusters, instead of breaking the core into similarly sized blocks. Other possibility could be the existence of overlapping communities, which would "blur" the division between clusters for non-overlapping clustering algorithms, such as the ones used in our experiments. In both cases, as granularity gets finer, there would be a tendency of having greater numbers of very small, even unitary, clusters, all the while having a few large ones. This behavior is observable in our results, but we cannot guarantee that one of those situations is the cause of this behavior, as our evaluation only uses edge counts, and that kind of characteristic would take into account stronger structural characteristics.

When looking at the results obtained for the Yeast biological dataset, presented in Figure 6.13, most clusters found are not very externally sparse. It is easy to see that, for all algorithms tested, most of the clusters found have conductance scores higher than 0.5, which means half of their total edges connect their vertices to other clusters. As for their internal densities, as discussed previously, the results show a slight tendency to have clusters that are either very dense or very sparse. Nevertheless, a larger part of the clusters found are on the internally sparser spectrum of the results, no matter the clustering algorithm used. Because of the way our internal density index works, we know that there exists similarly sized clusters which are denser than those very sparse clusters found; they just were not found by the clustering techniques used. However, since we don't know anything else about those denser clusters possible, *i.e.*, their external density, we can't affirm if they are globally better than the ones found by those clustering algorithms. On the other hand, the clusters found were, in general, not good neither internally nor externally, which raises doubts about the effectiveness of those algorithms when applied to this kind of network. The same behavior can be observed in the C. Elegans neural network results, which are not shown for brevity.

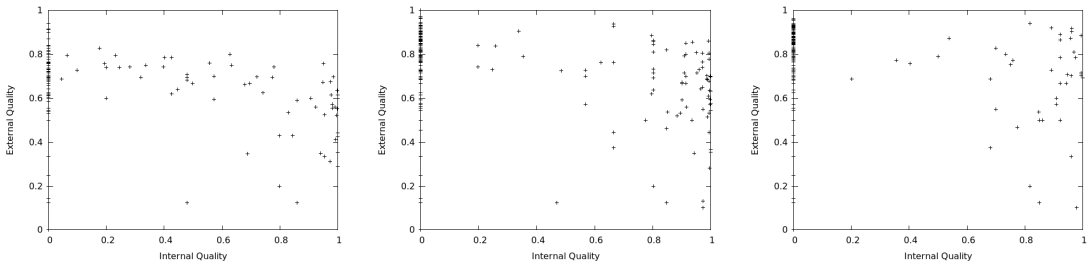
As for the technological networks, they show different kinds of results, depending on the network itself, and not only its origin. The results for the Power Grid network, presented on Figure 6.14, have internal density scores that are well spread through the spectrum of possible IDI values. However, this behavior is not coherent with the ones from the Gnutella networks studied. Figure 6.15 presents only the results for the 30/08/2002 Gnutella network snapshot, but its results are representative of both snapshots. From the Gnutella scores, it is easy to identify a trend of clusters being either very good or very bad IDI-wise, with very few clusters getting scores in-between those two extremes.



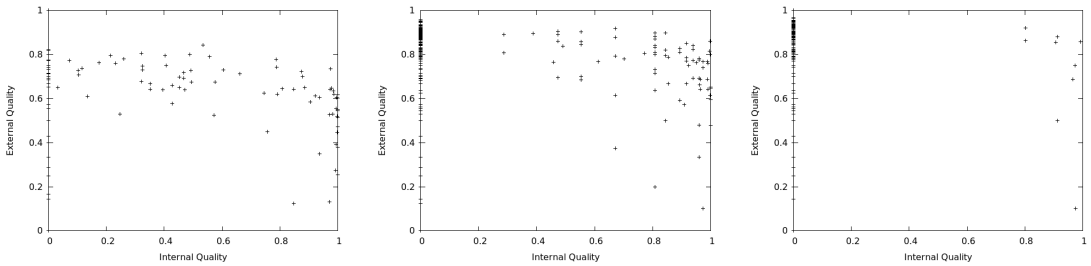
(a) Cluto ($k = 167$) (b) Cluto ($k = 521$) (c) Cluto ($k = 910$)



(d) Graclus ($k = 167$) (e) Graclus ($k = 521$) (f) Graclus ($k = 910$)



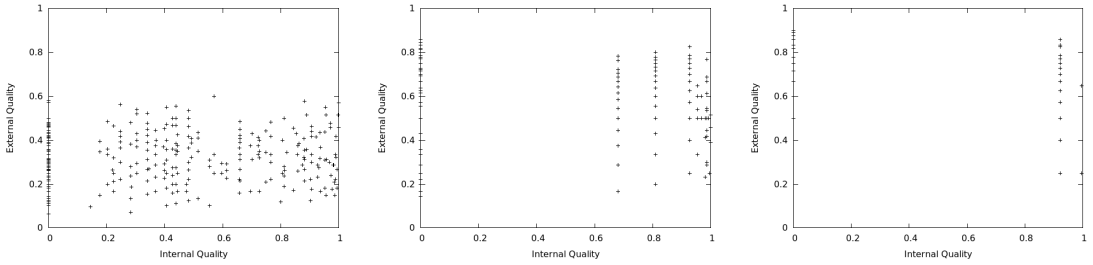
(g) MCL ($i = 1.4$) (h) MCL ($i = 2$) (i) MCL ($i = 4$)



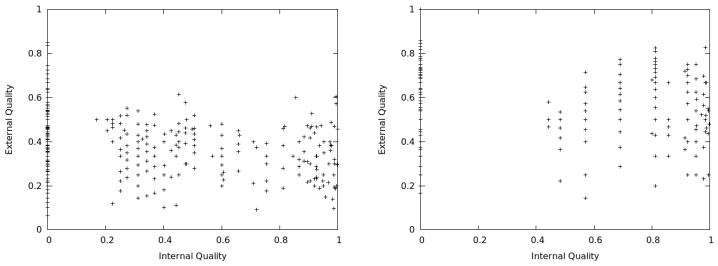
(j) SCPS ($k = 167$) (k) SCPS ($k = 521$) (l) SCPS ($k = 910$)

Figure 6.13: Internal vs. external quality values for clusters from the Yeast dataset.

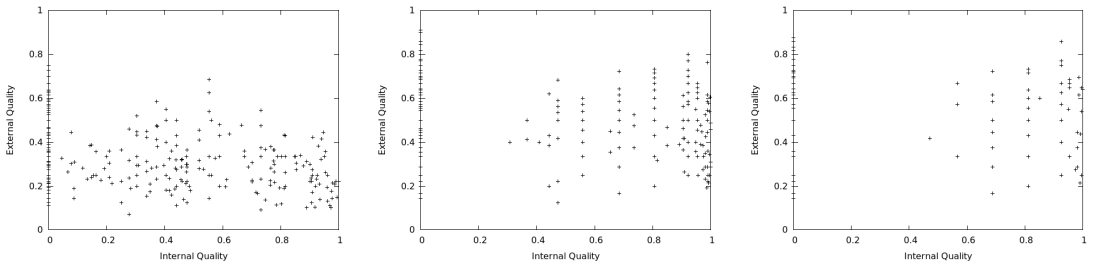
But what causes this difference in scoring behavior? This could happen because of extreme sparseness, a characteristic of Gnutella networks, causing abnormal behavior in the IDI evaluation process, but the Power Grid network, with roughly 1.33 edges for each vertex, is sparser than the Gnutella snapshots studied, with 2.41 and 2.36 edges per vertex, respectively. One probable cause for this behavior might be the way those edges are distributed through the graph. Even for the roughest granularity tested



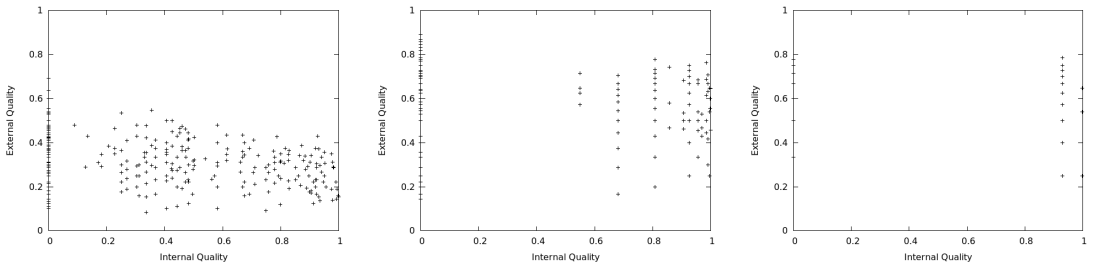
(a) Cluto ($k = 422$) (b) Cluto ($k = 1597$) (c) Cluto ($k = 2892$)



(d) Graclus ($k = 422$) (e) Graclus ($k = 1597$)



(f) MCL ($i = 1.4$) (g) MCL ($i = 2$) (h) MCL ($i = 4$)

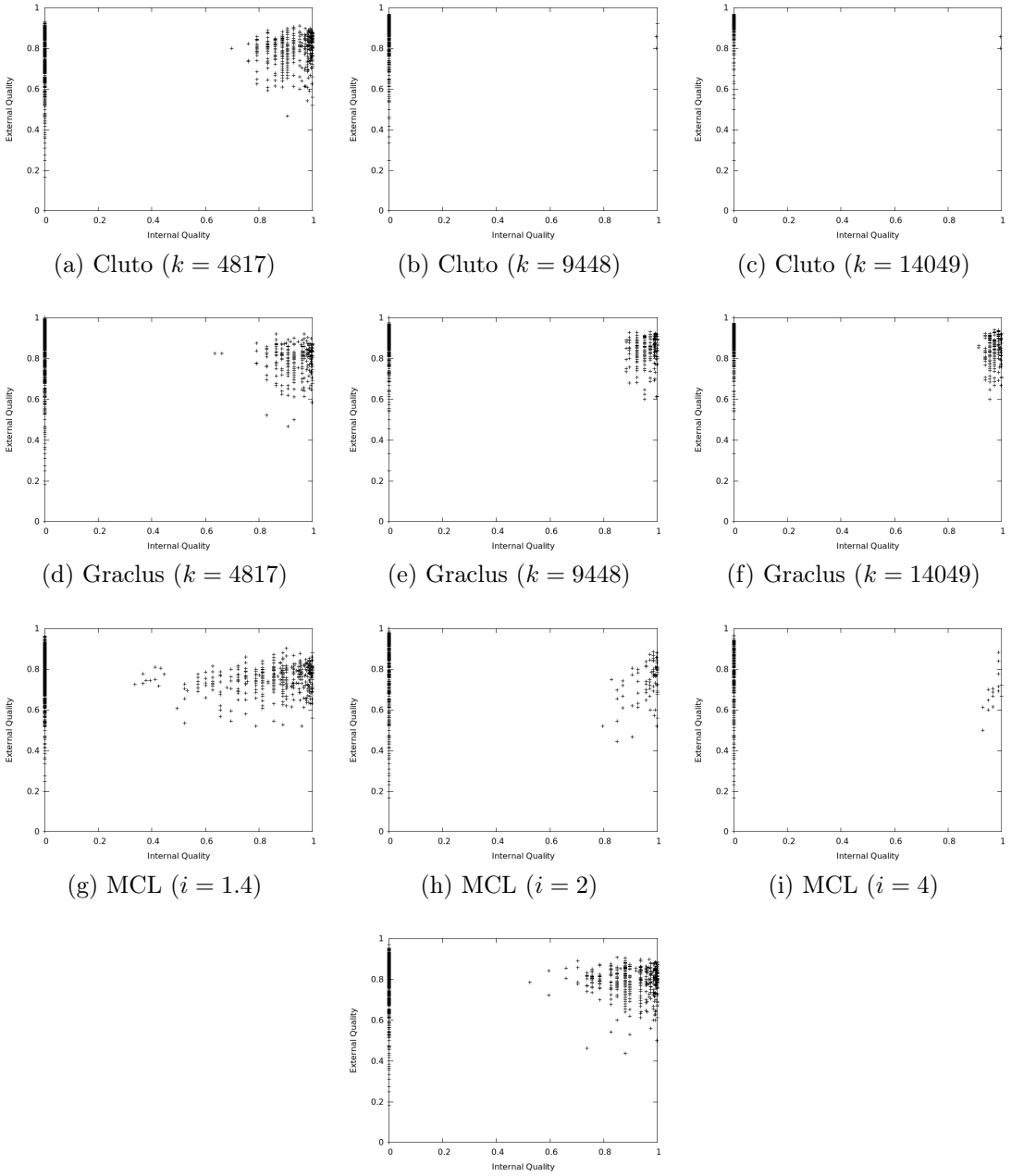


(i) SCPS ($k = 422$) (j) SCPS ($k = 1597$) (k) SCPS ($k = 2892$)

Figure 6.14: Internal vs. external quality values for clusters from the Power Grid dataset.

(MCL with $i = 1.4$), the Gnutella networks have smaller clusters, with an average of 7.6 and 7.3 vertices by cluster, against 11.8 for the same clustering configuration of the Power Grid network.

The excess of clusters found on the Gnutella Networks caused them to be very small and very sparse, since having more clusters, in general, means that more external



(a) Cluto ($k = 4817$) (b) Cluto ($k = 9448$) (c) Cluto ($k = 14049$)

(d) Graclus ($k = 4817$) (e) Graclus ($k = 9448$) (f) Graclus ($k = 14049$)

(g) MCL ($i = 1.4$) (h) MCL ($i = 2$) (i) MCL ($i = 4$)

(j) SCPS ($k = 4817$)

Figure 6.15: Internal vs. external quality values for clusters from the Gnutella P2P network (30/08/02) dataset.

edges will exist. Also, the structure of a Gnutella network, mostly composed by star-like formations of leaf nodes connected to one, and only one, super-node, has the tendency to make the smaller possible connected subgraphs to be, in general, trees. That causes most small clusters that are even slightly denser than a tree to be considered dense, according to our proposed method for internal density evaluation. This is a degener-

ate case, brought out by the kind of cluster and granularity found by the clustering algorithms used, but it should be better handled by our metric nonetheless. Variations on the parameter D of the IDI, the use of different weights for different cluster size or clustering granularity might be possible solutions for this kind of situation.

6.3 Final Remarks

In this chapter, we have shown the results from experiments done in order to test the performance of our proposed quality metric, specially when compared to other popular ones. We saw that our metric presented strong disagreements with the results from other metrics, and by looking at the details of some of those cases, we saw that our metric gave scores that were more closely related to the canonical representation of a good cluster, structurally-wise. Also, we saw that our metric did not present strong biases, for or against, clusters based only on their size. This is true for neither silhouette nor for modularity.

Chapter 7

Conclusions

Quality evaluation metrics, an essential part of the graph clustering problem, does not receive the necessary attention. Many quality metrics were proposed in the literature, but few authors went beyond the most simple kinds of validation and testing procedures. So, there is no consensus on the existence of one quality metric that performs better than the others, or even if they correctly evaluate cluster quality in more complex scenarios.

Because of that, we have studied some of the most popular quality metrics available in the literature. We compared their scoring behaviors for clusters obtained from real world graphs of different sizes and origins, using different clustering strategies and granularity levels. We discovered that those quality metrics present strong biases that were consistent for different types of graphs and clustering algorithms. Those biases caused some of the quality metrics considered to always favor results with fewer clusters, while others favored clusterings with more clusters. This is problematic, as those biases had no correlation to the kind of structure expected from a well-formed cluster. Also, this kind of behavior was more easily seen on the larger graphs than in the smaller ones. Since those smaller clusters have known optimal clusterings and, because of that, were the ones generally used to validate the efficacy of those metrics, this casts doubts on the ability of those metrics to correctly evaluate cluster quality.

Based on that, we evaluated some of those popular quality metrics even further, in order to discover the reason that caused them to present such undesirably biased scores. Our conclusion was that those quality metrics did not correctly evaluate the internal density of clusters, one of the two key aspects that define the structure of a well-formed cluster, the other being inter-cluster sparsity. Another problem we have detected was that those metrics used the same standards to evaluate clusters from graphs with wildly different structural characteristics, such as social and technological

ones. This causes clusterings from naturally sparser graphs to be severely penalized in their quality scores, making it hard to compare results from them.

To solve this problem, we proposed a novel method to evaluate internal cluster density. To do so, it uses the expected edge count from connected subgraphs of the same size as the cluster, induced from the graph being studied. This way, not only are we using both vertex and edge information to evaluate internal density, giving a more complete view of the clusters density, but we are also using the graph itself to determine the threshold that identifies what is dense and sparse in the context at hand. Through the aggregation of this internal quality index and Conductance, an external sparsity index, we obtained a new quality evaluation metric that provided a better view of a cluster's structural quality. When compared to results from the classical clustering quality evaluation metrics, we showed that our metric was capable to correctly penalize badly formed clusters that would nevertheless be well regarded by those metrics, while at the same time giving high scores to clusters that presented good structural characteristics.

7.1 Future Work

Even though our quality metric presents some nice improvements over other quality metrics from the literature, it is not perfect. It still has some room from improvement, as follows.

The first problem of our metric is that smaller clusters have a greater tendency of presenting high IDI values than larger ones. This is only natural, as smaller clusters need significantly fewer internal edges to become closer to a clique structure of the same size, which is the best structure possible, internal density-wise. Since complex networks are naturally sparse, the larger the cluster, the smaller the chance of it being very dense. However, since it is harder to have such dense and large cluster, when that happens, it might be fair to give it a higher overall rating. More testing to evaluate this possibility is necessary, though, to properly evaluate the effect of such approach.

Another problem regarding our metric is that, sometimes, being denser than expected might still be too sparse. Consider a graph with k vertices and k edges, where k is a fairly large number. In this case, most clusters of a given size s will be trees, since this graph is not a tree only because of one extra edge, but clusters might possess that extra edge. So, for IDI, those clusters with the extra edge will have very high scores because of the graph's structure, even though they are not well structured. Of course, this case is very specific, but it is important to identify the

impact of this kind of behavior. For example, for larger clusters, specially approaching the size of the graph itself, most clusters of the same size will be mainly equal, with just a few different vertices among them. Since they are almost the same, the variability of expected internal edge counts will be small, causing a problem similar to the one described earlier. More work would be necessary to determine if that kind of limitation on the cluster sizes would be enough to avoid this kind of problem.

A third problem is that the sampling process used to evaluate the expected values of cluster density is computationally costly. Considering that the other metrics evaluated are also expensive, requiring large matrix multiplications, all pairs shortest paths or min-cut max-flow computations, this is not one of our metrics' biggest flaw. Nevertheless, finding a way to decrease, or even outright remove the necessity for samplings would be highly positive. If there is a way to model the curve that describes the distribution of internal densities and uses characteristics from the target graph as parameters to do so, than this might be possible.

7.2 Published Articles

Part of the results presented here have already been published. The evaluation of currently used quality metrics and detection of their biases was presented in the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) [Almeida et al., 2011]. The in-depth study of those quality metrics' biases and proposal of a new method for internal cluster density evaluation was published in the Journal of Information and Data Management (JIDM) [Almeida et al., 2012].

Bibliography

Abrahao, B., Soundarajan, S., Hopcroft, J., and Kleinberg, R. (2012). On the separability of structural classes of communities. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 624--632, New York, NY, USA. ACM.

Almeida, H., Guedes, D., Meira, W., and Zaki, M. J. (2011). Is there a best quality metric for graph clusters? In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECML PKDD'11, pages 44--59, Berlin, Heidelberg. Springer-Verlag.

Almeida, H., Neto, D. O. G., Jr., W. M., and Zaki, M. J. (2012). Towards a better quality metric for graph cluster evaluation. 3(3):378–393.

Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149--11152.

Bagrow, J. P. and Bollt, E. M. (2005). Local method for detecting communities. *Phys. Rev. E*, 72(4):046108.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Berman, H. (2012). Stat trek. Online; accessed: 15-May-2012.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008a). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20:172–188.

Brandes, U., Gaertler, M., and Wagner, D. (2003). Experiments on graph clustering algorithms. In *In 11th Europ. Symp. Algorithms*, pages 568--579. Springer-Verlag.

Brandes, U., Gaertler, M., and Wagner, D. (2008b). Engineering graph clustering: Models and experimental evaluation. *J. Exp. Algorithmics*, 12:1--26.

Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., and Chen, R. (2003). Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450.

Chakrabarti, D. and Faloutsos, C. (2006). Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2.

Clauset, A. and Moore, C. (2005). Accuracy and scaling phenomena in internet mapping. *Phys Rev Lett*, 94(1):018701. automatic medline import.

Danon, L., Díaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.

Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944--1957.

Dongen, S. M. (2000). *Graph Clustering by Flow Simulation.* PhD thesis, University of Utrecht, The Netherlands.

Dongen, S. V. (2008). Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141.

Du, N., Wu, B., Pei, X., Wang, B., and Xu, L. (2007). Community detection in large-scale social networks. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 16--25, New York, NY, USA. ACM.

Ebel, H., Mielsch, L.-I., and Bornholdt, S. (2002). Scale-free topology of e-mail networks.

Egghe, L. and Rousseau, R. (1990). *Introduction to Infometrics.* Elsevier Science Publishers, Amsterdam.

Erdös, P. and Rényi, A. (1959). On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290--297.

Faloutsos, C. (2010). Invited talk: Acm sig-kdd innovations award 2010 acceptance speech. ACM Conference on Knowledge Discovery and Data Mining (SIGKDD).

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826.

Gleich, D. F. and Seshadhri, C. (2012). 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. ACM.

Good, B. H., de Montjoye, Y. A., and Clauset, A. (2010). Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106+.

Guha, S. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35--58.

Gustafson, M., Hörnquist, M., and Lombardi, A. (2006). Comparison and validation of community structures in complex networks. *Physica A: Statistical Mechanics and its Application*, 367(1):559--576.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100--108.

Huang, Zhexue (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values.

Huberman, B. A. (2001). *The Laws of the Web*. MIT Press, Cambridge, MA.

Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569--4574.

J., C. W. (1980). *Practical Non-Parametric Statistics*. John Wiley and Sons, New York, NY, USA.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651--654.

Jiang, J. Q., Dress, A. W., and Yang, G. (2009). A spectral clustering-based framework for detecting community structures in complex networks. *Applied Mathematics Letters*, 22(9):1479 – 1482.

Kannan, R., Vempala, S., and Vetta, A. (2004). On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497--515.

Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359--392.

King, Przulj, N., and Jurisica, I. (2004). Protein complex prediction via cost-based clustering. *Bioinformatics (Oxford, England)*, 20(17):3013--20.

Knuth, D. E. (1993). *The Stanford GraphBase: a platform for combinatorial computing*. ACM, New York, NY, USA.

Krishnamurthy, B. and Wang, J. (2000). On network-aware clustering of web clients. *SIGCOMM Comput. Commun. Rev.*, 30:97--110.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the web for emerging cyber-communities. *Comput. Netw.*, 31:1481--1493.

Laboratories, S. U. S. E., Cover, T., Army, U. S., Navy, U. S., and Force, U. S. A. (1966). *Estimation by the nearest neighbor rule*. Technical report. Systems Theory Laboratory, Stanford Electronics Laboratories, Stanford University.

Lancichinetti, A. and Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117.

Lappas, T., Liu, K., and Terzi, E. (2009). Finding a team of experts in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 467--476, New York, NY, USA. ACM.

Leicht, E. A. and Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters*, 100:118703.

Leskovec, J. (2012). Snap network analysis library. Online; accessed: 15-May-2012.

Leskovec, J. and Faloutsos, C. (2006). Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 631--636, New York, NY, USA. ACM.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2008). Statistical properties of community structure in large social and information networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 695--704, New York, NY, USA. ACM.

Leskovec, J., Lang, K. J., and Mahoney, M. (2010). Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 631--640, New York, NY, USA. ACM.

Lohr, S. L. (2010). *Sampling: Design and Analysis*, chapter Simple Probability Samples, pages 25--72. Brooks/Cole, Boston, USA, 2 edition.

Lu, Q., Korniss, G., and Szymanski, B. (2009). The naming game in social networks: community formation and consensus engineering. *Journal of Economic Interaction and Coordination*, 4(2):221–235.

Meilă, M. (2007). Clustering by weighted cuts in directed graphs. In *SIAM International Conference on Data Mining*, pages 135--144.

Nascimento, M. C. and de Carvalho, A. C. (2011). Spectral methods for graph clustering – a survey. *European Journal of Operational Research*, 211(2):221 – 231.

Nepusz, T. and Bazso, F. (2007). Likelihood-based clustering of directed graphs. pages 189 –194.

Nepusz, T., Sasidharan, R., and Paccanaro, A. (2010). Scps: a fast implementation of a spectral method for detecting protein families on a genome-wide scale. *BMC Bioinformatics*, 11(1):120.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2).

Newman, M. E. J. (2003a). Fast algorithm for detecting community structure in networks.

Newman, M. E. J. (2003b). Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126.

Newman, M. E. J. (2003c). The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167--256.

Newman, M. E. J. (2009). Random graphs with clustering. *Phys. Rev. Lett.*, 103:058701.

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

Pan, Y., Li, D.-H., Liu, J.-G., and Liang, J.-Z. (2010). Detecting community structure in complex networks via node similarity. *Physica A: Statistical Mechanics and its Applications*, 389(14):2849 – 2857.

Pereira-Leal, J. B., Enright, A. J., and Ouzounis, C. A. (2004). Detection of functional modules from protein interaction networks. *Proteins: Structure, Function, and Bioinformatics*, 54(1):49--57.

Reddy, K. P., Kitsuregawa, M., Sreekanth, P., and Rao, S. S. (2002). A Graph Based Approach to Extract a Neighborhood Customer Community for Collaborative Filtering. In *DNIS '02: Proceedings of the Second International Workshop on Databases in Networked Information Systems*, pages 188--200, London, UK. Springer-Verlag.

Ribeiro, B. and Towsley, D. (2010). Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 390--403, New York, NY, USA. ACM.

Sarkar, P. and Moore, A. W. (2010). Fast nearest-neighbor search in disk-resident graphs. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 513--522, New York, NY, USA. ACM.

Satuluri, V. and Parthasarathy, S. (2009). Scalable graph clustering using stochastic flows: applications to community discovery. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737--746, New York, NY, USA. ACM.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27--64.

Shamir, R., Sharan, R., and Tsur, D. (2004). Cluster graph modification problems. *Discrete Appl. Math.*, 144(1-2):173--182.

Shyam Boriah, V. C. and Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In *Proceedings of 2008 SIAM Data Mining Conference*.

Šíma, J. and Schaeffer, S. E. (2006). On the NP-completeness of some graph cluster measures. In Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., and Štuller, J., editors, *Proceedings of the Thirty-second International Conference on Current Trends in Theory and Practice of Computer Science (Sofsem 06)*, volume 3831 of *Lecture Notes in Computer Science*, pages 530--537, Berlin/Heidelberg, Germany. Springer-Verlag GmbH.

Smith, R. D. (2002). Instant messaging as a scale-free network.

Solomonoff, R. and Rapoport, A. (1951). Connectivity of random nets. *Bulletin of Mathematical Biology*, 13(2):107--117.

Steinbach, M., Karypis, G., and Kumar, V. (2000a). A comparison of document clustering techniques. In Grobelnik, M., Mladenic, D., and Milic-Frayling, N., editors, *KDD-2000 Workshop on Text Mining, August 20*, pages 109–111, Boston, MA.

Steinbach, M., Karypis, G., and Kumar, V. (2000b). A comparison of document clustering techniques.

Steinhaeuser, K. and Chawla, N. V. (2010). Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31(5):413 – 421.

Sudipto Guha, R. R. and Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*, page 512, Washington, DC, USA. IEEE Computer Society.

Tan, P.-N., Kumar, V., and Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 32--41, New York, NY, USA. ACM.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Travers, J. and Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, 32(4):425--443.

Watts, D. J. and Strogatz, S. H. (1998a). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440--442.

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

Wong, W. and Fu, A. (2000). Incremental Document Clustering for Web Page Classification.

Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. A. J. (2007). Scan: a structural clustering algorithm for networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824--833, New York, NY, USA. ACM.

Xu, Y., Olman, V., and Xu, D. (2002). Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18(4):536--545.

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

Zaidi, F., Archambault, D., and Melançon, G. (2010). Evaluating the quality of clustering algorithms using cluster path lengths. In *Proceedings of the 10th industrial conference on Advances in data mining: applications and theoretical aspects*, ICDM'10, pages 42--56, Berlin, Heidelberg. Springer-Verlag.

Zhang, Tian, Ramakrishnan, Raghu, and Livny, Miron (1996). BIRCH: an efficient data clustering method for very large databases. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103--114, New York, NY, USA. ACM.

Zhou, D., Huang, J., and Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1036--1043, New York, NY, USA. ACM.

Zhou, Y., Cheng, H., and Yu, J. X. (2009). Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.*, 2(1):718--729.