UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Redes Complexas para a Ciência da Computação

Detecção de Comunidades em Redes Complexas

Glenda Michele Botelho

Fabiano Berardo de Sousa

Junho de 2011

Resumo

A detecção de comunidades em redes complexas é muito importante, pois permite a extração de informações relevantes das redes analisadas. Este trabalho tem como objetivo principal apresentar uma revisão bibliográfica sobre detecção de comunidades, destacando diversos algoritmos. Inicia-se com conceitos introdutórios referentes à presença de estruturas de comunidades em redes e noções sobre *Clustering* em Aprendizado de Máquina, devido à correlação entre as técnicas de ambas as áreas. Em seguida, destacam-se pesquisas mais recentes focadas na detecção de comunidades como, por exemplo, o desenvolvimento da *Medida de Modularidade* [Newman and Girvan 2004] e propostas posteriores para seu refinamento, como a *Otimização Extrema* [Duch and Arenas 2005] e a *Otimização utilizando o método de Monte Carlo com Simulated Annealing e Basin Hopping* [Massen and Doye 2005], além de métodos Espectrais [Newman 2006] e Locais [Muff et al. 2005]. Por fim, apresentam-se algumas conclusões referentes aos métodos de detecção, ressaltando as suas principais limitações.

Sumário

Lista de Figuras

Capítulo 1

Introdução

Redes Complexas são grafos que apresentam uma estrutura topográfica não trivial, compostos por um conjunto de ítens, chamados de vértices ou nós, cujas conexões entre eles são chamadas de arestas. É uma área multidisciplinar iminente cuja pesquisa iniciou-se em meados de 1930 no ramo da sociologia, quando redes foram utilizadas para modelar e analisar o comportamento da sociedade e as relações entre indivíduos. Atualmente, a fim de resolver problemas específicos de diversas áreas, as redes podem ser utilizadas para representar e estudar diversos aspectos do mundo real, tais como a estrutura física de grandes redes de computadores (Internet e *World Wide Web*), redes sociais, redes neurais biológicas, redes metabólicas, cadeias alimentares, entre outros. A Figura 1.1 apresenta a Internet modelada como rede complexa.

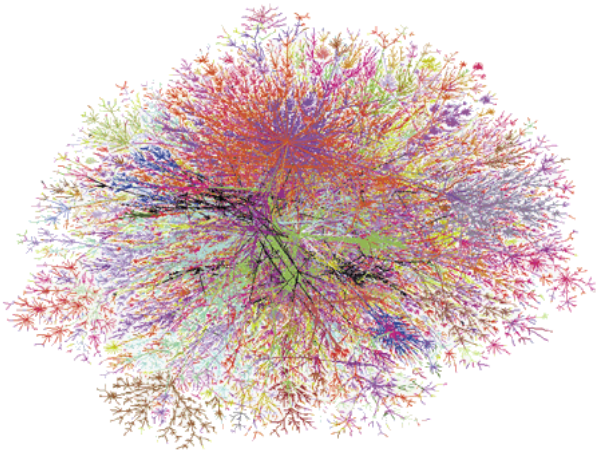

Figura 1.1: Visualização da estrutura da rede Internet. Fonte: http://www.research.ibm.com/nips03workshop/, último acesso em 18/05/2011.

Inicialmente, as pesquisas relacionadas às redes complexas eram baseadas em medidas como centralidade (vértice mais central) e conectividade (vértice com maior número de conexões).

No entanto, o avanço da tecnologia e o aumento do poder computacional permitiram a análise de grandes redes e as pesquisas passaram a considerar propriedades estatísticas em grande escala (milhões ou bilhões de vértices). Tal mudança revelou características que diferem substancialmente as redes do mundo real das redes aleatórias como, por exemplo, a presença de propriedades organizacionais bastante robustas e distintas nas redes reais. Com isso, pôde-se perceber que a estrutura das redes complexas não seguem um padrão regular e apresenta características próprias, as quais revelam como as redes são formadas e como suas estruturas podem ser exploradas na análise de um determinado problema.

Uma característica importante das redes complexas é a presença de estruturas de comunidades, ou seja, grupos nos quais os vértices são mais intensamente conectados entre si do que com o restante da rede, conforme mostra a Figura 1.2. A detecção das comunidades juntamente com a extração de conhecimento de sua estrutura vem sendo bastante explorada em pesquisas de aprendizado de máquina e mineração de dados [Stauffer et al. 2003]. Ressalta-se que a detecção de comunidades ainda é um dos grandes desafios da área de aprendizado de máquina, pois a maioria dos algoritmos tradicionais apresentam-se computacionalmente instáveis para tratar quantidades tão grandes de vértices e arestas, fato que torna esta área de pesquisa relevante e promissora [Newman 2004, Newman and Girvan 2004, Oliveira et al. 2008, Quiles et al. 2008].

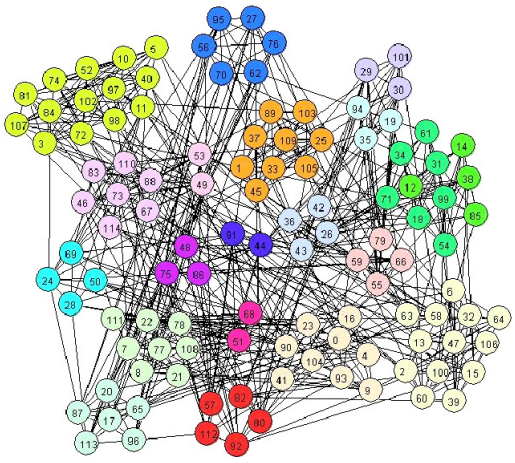

Figura 1.2: Exemplo de comunidades presentes em uma rede. Figura adaptada de Newman [Newman and Girvan 2004].

A detecção de comunidades tem correlação com as técnicas de *clustering* pertencentes à área de aprendizado de máquina. Essa correlação possibilita que algoritmos implementados para uma área possam ser facilmente adaptados para outra e vice-versa. Muitas vezes utilizam-se métodos de particionamento para a obtenção de subgrafos que representam individualmente cada comunidade presente na rede. No entanto, apenas métodos de particionamento não são

suficientes, visto que não se conhece *a priori* "se"e "como"a rede separa os vértices em comunidades, nem o número e o tamanho das comunidades existentes. Dessa forma, o *clustering* hierárquico pode ser usado para descobrir divisões naturais da rede.

Algoritmos de *clustering* hierárquico baseiam-se em conceitos de similaridade entre os vértices e são classificados em duas abordagens: aglomerativa e divisiva [Jain and Dubes 1988]. Na abordagem aglomerativa cada vértice da rede é considerado uma comunidade unitária. Em seguida, arestas são iterativamente adicionadas ao grafo para unir os subgrafos até que todos os vértices pertençam a apenas um grafo. Já a abordagem divisiva inicia com apenas um grafo contendo todos os vértices e procede dividindo este grafo em subgrafos cada vez menores, até que cada vértice seja um grafo isolado ou até que se alcance algum critério de parada como, por exemplo, o número de subgrafos desejados.

Além das abordagens baseadas em *clustering* hierárquico, outras pesquisas apresentam algoritmos que, alternativamente, não se baseiam em medidas de similaridade como, por exemplo, a medida de *Betweenness*, proposta por Girvan e Newman [Girvan and Newman 2002], a qual utiliza o cálculo do caminho mínimo entre os vértices para fundamentar a detecção de comunidades. Posteriormente, o próprio Newman [Newman 2004] propôs uma nova abordagem, a qual chamou de *Medida de Modularidade*, para mensurar a qualidade de possíveis divisões na rede, sem a necessidade do conhecimento prévio da estrutura da mesma. A partir deste trabalho de Newman, várias pesquisas têm sido realizadas a fim de otimizar a função de modularidade, como a otimização extrema de Duch e Arenas [Duch and Arenas 2005] e a otimização utilizando o método de Monte Carlo proposta por Massen e Doye [Massen and Doye 2005].

Assim sendo, este trabalho tem como objetivo realizar um estudo bibliográfico sobre detecção de comunidades, desde seu início até as mais recentes e diversas abordagens presentes na literatura. O Capítulo 2 tece uma rápida introdução ao conceito de *Clustering* em Aprendizado de Máquina como forma de fundamentar os conceitos apresentados no capítulo seguinte. O Capítulo 3 apresenta, portanto, as considerações iniciais sobre detecção de comunidades em redes complexas e, em seguida, aborda diversos métodos e medidas como a Centralidade *Betweenness*, a Medida de Modularidade e suas Otimizações, além de outros algoritmos, subdivididos, para melhor efeito didático, em categorias como Métodos Espectrais e Locais. O Capítulo 4 discorre sobre aplicações e desafios da área. Por fim, o Capítulo 5 apresenta as considerações finais.

Capítulo 2

Noções sobre *Clustering*

Conforme visto no Capítulo 1, algoritmos para detecção de comunidades correlacionam-se com métodos para detecção de agrupamentos (*Clustering*) utilizados em Aprendizado de Máquina, mais precisamente com técnicas de aprendizado não supervisionado. Diferentemente do processo de classificação, no qual um classificador é treinado, sob o olhar de um supervisor, para encontrar as categorias presentes nos dados, a detecção de agrupamento tenta identificar, sem o auxílio de um supervisor, subdivisões naturais em um conjunto de dados, nos quais os elementos não possuem um atributo meta (classe). Esse processo de aprendizado considera propriedades intrínsecas presentes nos elementos como forma de estabelecer similaridade ou dissimilaridade, formando agrupamentos com a maior homogeneidade possível entre os elementos.

Este capítulo apresenta uma rápida introdução ao conceito de *Clustering*, abordando os principais modelos de algoritmos, conforme a definição de Jain e Dubes [Jain and Dubes 1988]: algoritmos particionais e hierárquicos, além de também mencionar os algoritmos baseados em densidade. A Seção 2.1 descreve os algoritmos particionais e cita um de seus principais exemplos: o algoritmo *K-means*. A seção 2.2 discorre sobre algoritmos baseados em densidade e tece comentários sobre um dos primeiros trabalhos da área, o *DBSCAN*. Por fim, algoritmos hierárquicos são descritos na Seção 2.3. Estes últimos são os principais modelos de algoritmos nos quais trabalhos sobre detecção de comunidades em redes complexas se baseiam, conforme será visto no Capítulo 3.

2.1 Algoritmos Particionais

Algoritmos particionais constroem uma partição D de n objetos dentro de um conjunto de K grupos (*clusters*). Tipicamente, o processo se inicia com uma partição inicial de D e usa

uma estratégia iterativa para otimizar uma função objetivo. Cada agrupamento é representado por um "centro de gravidade"ou *centróide* (no caso do algoritmo *K-Means*, discutido com mais detalhes na subseção seguinte), ou por um dos objetos do grupo que mais se aproxima de seu centro (como no *K- Medoid*, uma variação do *K-Means*, que utiliza a mediana ao invés da média, como medida de centro).

Para tanto, estes algoritmos operam em dois estágios: No primeiro, determinam k representantes para os K agrupamentos que se deseja encontrar (ou seja, K "centros de gravidade", um para cada agrupamento) de forma a minimizar a função objetivo. No segundo, atribuem cada objeto ao agrupamento cujo representante estiver mais próximo. Normalmente, utiliza-se a distância euclidiana como métrica para avaliação da proximidade, porém outras medidas podem ser utilizadas [Ester et al. 1996].

2.1.1 K-Means

O *K-Means* é um algoritmo particional que se baseia no cálculo do *Erro Quadrático* como sua função objetivo. O objetivo do algoritmo é, portanto, obter uma partição que minimiza o erro quadrático para um número K fixo de agrupamentos (K deve ser fornecido pelo usuário). O erro quadrático pode ser expresso como segue:

$$E = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \bar{x}^{(j)})^2 \tag{2.1}$$

onde $d(x_i, \bar{x}^{(j)})$ representa a distância euclidiana entre um elemento x_i (pertencente ao cluster C_j) e o centróide do cluster C_j, representado por $\bar{x}^{(j)}$.

O algoritmo clássico começa inicializando K centróides para os K agrupamentos que se deseja encontrar. Em seguida, cada elemento do conjunto de dados é associado ao agrupamento cujo centróide está mais próximo. Na etapa seguinte, o centróide é recalculado de forma a ocupar a média das coordenadas dos elementos atribuídos àquele agrupamento. O processo continua até que os centróides não sejam mais alterados.

A complexidade do algoritmo é $O(n)$, uma vez que o número de iterações é pequeno e $K << n$. Porém, apesar da baixa complexidade (fato que resultaria em relativo baixo custo computacional se comparados a outros tipos de algoritmos), algoritmos particionais exigem que se estabeleça *a priori* o número de agrupamentos que se deseja identificar, o que os tornam pouco adequados para tratarem problemas relacionados à detecção de comunidades em redes complexas, uma vez que a descoberta do número de comunidades e como tais comunidades se relacionam permitem extrair conhecimento sobre propriedades e relações muitas vezes

desconhecidas nestes sistemas.

2.2 Algoritmos Baseados em Densidade

Este tipo de algoritmo considera que um agrupamento é uma região com alta densidade de objetos e, assim sendo, tenta identificar regiões altamente densas separadas por regiões com baixa densidade, tornando-o capaz de detectar agrupamentos de diversos formatos. A subseção seguinte descreve o algoritmo DBSCAN, um dos primeiros trabalhos da área.

2.2.1 DBSCAN

O algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) foi desenhado para descobrir tanto agrupamentos como também ruídos e *outliers* presentes nos dados. A forma como ele trabalha é baseado nas definições seguintes [Ester et al. 1996]:

- $Def.1$: A ε-vizinhança de um objeto p, denotado por $N_\varepsilon(p)$, é definida como sendo o número total de objetos que permanecem dentro do raio ε: $N_\varepsilon(p) = \{q \in D \mid dist(p,q) \leq q\}$, onde D representa a base de objetos.

- $Def.2$: Um objeto p é dito *densamente alcançável de forma direta* por um ponto q, se: i) $p \in N_\varepsilon(q)$; ii) $|N_\varepsilon(q)| \geq \mu$, onde μ é o número mínimo de objetos que devem pertencenter a uma ε-vizinhança (neste caso, q é dito *núcleo*).

- $Def.3$: Um objeto p é dito *densamente alcançável* por um objeto q se existe uma cadeia de objetos $p_1, ..., p_n, p_1 = q, p_n = p$, tal que p_{i+1} é *densamente alcançável de forma direta* a partir de p_i.

- $Def.4$: Um objeto p é dito *densamente conectado* a um objeto q, se existe um objeto o tal que ambos, p e q são objetos *densamente alcançáveis* a partir de o.

- $Def.5$: Seja D uma base de objetos. Um cluster C é dito um conjunto não-vazio de D se: i) $\forall p, q$: se $p \in C$ e q é *densamente alcançável* a partir de p, então $q \in C$; ii) $\forall p, q \in C$: p é *densamente conectado* a q.

- $Def.6$: Seja $C_1, ..., C_k$ clusters da base de dados D. Defini-se *ruído* como o conjunto de objetos pertencentes à base D e que não pertencem a nenhum dos clusters $C_i, i = 1, ..., k$.

O algoritmo detecta *clusters* C_i descobrindo um de seus p *núcleos* e computando todos os objetos que são *densamente alcançáveis* a partir de p. Tais objetos *densamente alcançáveis*

são computados de forma iterativa a partir do cálculo de objetos *densamente alcançáveis de forma direta* a partir de p. O DBSCAN checa a ε-vizinhança de todo objeto p da base D. Se $N_\varepsilon(p)$ de um objeto p consiste em pelo menos μ objetos (como no caso de p ser um *núcleo*), um novo *cluster* C_i é criado. Em seguida, a ε-vizinhança de todo objeto $q \in C_i$ que ainda não foi processado é checada. Se q também for um *núcleo*, os vizinhos de q, os quais ainda não foram atribuídos ao *cluster* C_i, são adicionados ao *cluster* C_i e sua ε-vizinhança será checada no passo seguinte. Todo o processo é repetido até que não haja mais objetos a serem adicionados ao atual *cluster* C_i.

Um dos principais problemas do DBSCAN surge quando existe alta variação de densidade dentro de um *cluster*. Muitos trabalhos surgiram para corrigir e otimizar este e outros problemas. Outro algoritmo interessante baseado em densidade é o *Chameleon* [Karypis et al. 1999] que utiliza princípios de modelos dinâmicos e opera em duas fases: na primeira, o algoritmo gera um grafo de K-vizinhos mais próximos; na segunda, é utilizado um algoritmo hierárquico aglomerativo para encontrar um *cluster*, combinando, iterativamente, esses subgrafos. Algoritmos Hierárquicos serão vistos na seção seguinte.

2.3 Algoritmos Hierárquicos

Algoritmos hierárquicos geram uma sequência de partições aninhadas, a partir da partição D que contém a base de dados. Tal hierarquia é representada na forma de um *dendograma*, ou seja, uma árvore que lista a partição D e sua sequência de sub-partições. Esta sequência de partições aninhadas pode ser gerada a partir de duas abordagens: *divisiva* e *aglomerativa*. Na primeira, inicia-se com todos os objetos formando um grupo. Em seguida, subgrupos são gerados até que cada objeto da base seja considerado um grupo isoladamente. Na abordagem aglomerativa acontece o oposto: parte-se da situação em que cada objeto é considerado um grupo e, iterativamente, grupos são mesclados até que todos os objetos façam parte de um único grupo [Ester et al. 1996]. A Figura 2.1 ilustra um dendograma e suas respectivas abordagens divisiva e aglomerativa.

Os grupos são aglomerados ou divididos de acordo com uma métrica ou medida de similaridade qualquer (normalmente, utilizam-se distâncias). A vantagem deste tipo de algoritmo é sua flexibilidade em relação ao nível de granularidade dos agrupamentos que se deseja obter, os quais podem ser facilmente analisados pela representação gráfica proporcionada pelo dendograma. A partir da observação de um dendograma pode-se inferir qual divisão da partição D (número de sub-partições) é mais adequada ao contexto do problema. Por estes e outros motivos, algoritmos que seguem uma abordagem hierárquica são os mais utilizados em pesquisas

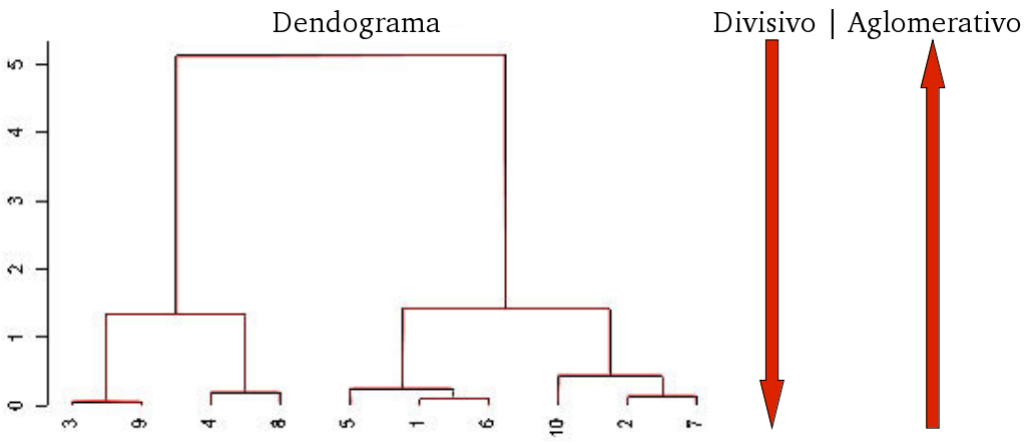

Figura 2.1: Dendograma que ilustra o funcionamento de algoritmos hierárquicos em uma base com 10 objetos. As setas indicam o sentido das abordagens divisiva e aglomerativa.

relacionadas com a detecção de comunidades em redes complexas, uma vez que a descoberta do número de comunidades e como tais comunidades se relacionam é parte do conhecimento a ser extraído dos sistemas que estas redes modelam. Estes conceitos, assim como diversas abordagens e algoritmos aplicados à detecção de comunidades serão vistos no capítulo seguinte.

Capítulo 3

Abordagens para a Detecção de Comunidades

3.1 Considerações Iniciais

Grande parte das redes reais apresentam estruturas de conexão não homogêneas caracterizadas pela presença de grupos nos quais os vértices são mais intensamente conectados entre si do que com o restante da rede, conforme pode ser observado na Figura 3.1, a qual apresenta uma rede com três comunidades bem definidas.

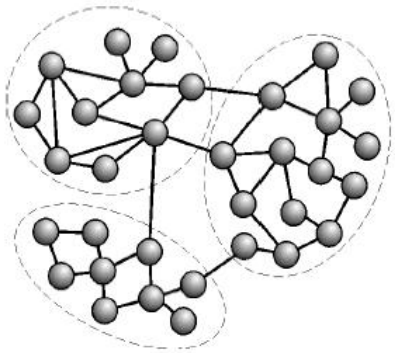

Figura 3.1: Rede com três comunidades bem definidas. Comunidades são grupos cujos vértices são mais intensamente interconectados. Figura adaptada de Costa [Costa et al. 2007].

A detecção de comunidades em grandes redes é uma tarefa útil pois vértices pertencentes à mesma comunidade tem mais chances de compartilharem propriedades e dinâmicas. Além disso, a quantidade e características das comunidades provê subsídios para identificar o tipo da rede, assim como entender sua organização e evolução dinâmica [Costa et al. 2007]. Por

exemplo, no caso da *World Wide Web*, páginas relacionadas ao mesmo assunto geralmente estão organizadas em comunidades, logo a identificação destas comunidades pode ajudar na tarefa de busca de informações.

Apesar da importância do conceito de comunidades, sua definição não é um consenso. Radicchi [Radicchi et al. 2004] propôs uma definição intuitiva baseada na comparação da densidade das arestas entre os vértices. Dessa forma, comunidades podem ser definidas no sentido forte ou fraco. No sentido forte, um subgrafo é uma comunidade se todos os seus vértices tem mais conexões entre eles que com o restante dos vértices da rede. Já no sentido fraco, um subgrafo é uma comunidade se a soma de todos os graus dos vértices dentro de um subgrafo é maior que daqueles que estão fora. No entanto, estas definições geram uma limitação: a união de comunidades também é uma comunidade. Para superar esta limitação, uma hierarquia entre as comunidades pode ser assumida *a priori* [Reichadt and Bornholdt 2006].

Outro problema fundamental relacionado à detecção de comunidades é como definir a melhor divisão da rede em suas comunidades constituintes, visto que em redes reais geralmente nenhuma informação está disponível sobre o número e tamanho das comunidades existentes. Com o objetivo de resolver este problema, Newman [Newman 2004] propôs uma medida, chamada de **Modularidade (Q)**, que mede a qualidade de uma divisão particular da rede. Com base nesta medida, pode-se estimar a qualidade do particionamento obtido por um determinando algoritmo de detecção de comunidades.

Existem vários algoritmos de detecção de comunidades. Alguns são relacionados com conceitos de ***clustering* hierárquico**, mais precisamente com os métodos **divisivos** (Seção 3.2) e **aglomerativos** (Seção 3.3), que geralmente se baseiam na similaridade entre os vértices para realizar a detecção de comunidades. Outros algoritmos não utilizam medidas de similaridade, dentre eles cita-se: os métodos **espectrais** (Seção 3.4), que se baseiam em autovalores e autovetores da matriz derivada da rede; os métodos **locais** (Seção 3.5), que avaliam a comunidade atual e suas comunidades vizinhas; a medida de ***Betwenness*** (Subseção 3.2.1), que se baseia no cálculo do caminho mínimo e a medida de **Modularidade** (Subseção 3.3.2), que estima a qualidade de divisões na rede.

A partir do desenvolvimento da medida de modularidade vários trabalhos foram propostos com o intuito de otimizá-la e obter um melhor particionamento da rede em comunidades. Dentre estes trabalhos têm-se a otimização extrema [Duch and Arenas 2005] e a otimização usando o método de Monte Carlo [Massen and Doye 2005], apresentados, respectivamente, nas subseções 3.6.1 e 3.6.2. A seções seguintes abordarão, com detalhes, estes e outros métodos.

3.2 Métodos Divisivos

Métodos divisivos buscam encontrar as arestas que conectam as diferentes comunidades e removê-las iterativamente, dividindo a rede em grupos desconexos de vértices, ou seja, inicia-se com um único grafo contendo todos os vértices e procede-se dividindo este grafo em sub-grafos cada vez menores. A seguir apresentam-se alguns algoritmos divisivos, os quais diferem de acordo com a medida usada para escolher a aresta que será removida.

3.2.1 Centralidade *Betweenness*

Um dos algoritmos divisivos mais populares foi proposto, em 2002, por Girvan e Newman [Girvan and Newman 2002] e utiliza o conceito de *Betweenness* para remover arestas que conectam comunidades. O *Betweenness* é uma medida usada para identificar arestas que conectam comunidades, apresentando valores altos para estas arestas e penalizando arestas que conectam vértices pertencentes a mesma comunidade. Para calcular o valor de *Betwenness* é necessário obter o caminho mínimo entre dois vértices, conforme apresentado na equação 3.1:

$$B_u = \sum_{ij} \frac{\sigma(i,u,j)}{\sigma(i,j)} \tag{3.1}$$

onde $\sigma(i,u,j)$ é o número de caminhos mínimos entre os vértices i e j que passam pelo vértice ou aresta u, $\sigma(i,j)$ é o número total de caminhos mínimos entre i e j e o somatório se aplica a todos os pares i e j de vértices distintos. Considerando a medida de *Betweenness* e uma rede com duas comunidades ligadas por um pequeno número de arestas, tem-se que todos os caminhos da rede com origem em um vértice de uma comunidade e destino em um vértice da outra comunidade devem passar por alguma destas arestas que conectam as comunidades. Desta forma, estas arestas terão alto valor de *Betweenness* e as arestas pertencentes à mesma comunidade terão um valor menor.

Segundo o algoritmo de Girvan e Newman [Girvan and Newman 2002], arestas com alto valor de *Betweenness* são removidas iterativamente. Depois de remover cada aresta, o *Betwenness* das arestas remanescentes deve ser recalculado. Com isso, a principal desvantagem do algoritmo é seu custo computacional. Em um grafo com M arestas e N vértices, a complexidade total para o cálculo do *Betwenness* é $O(M^2N)$.

3.2.2 Coeficiente de Agrupamento das Arestas

O algoritmo proposto por Radicchi [Radicchi et al. 2004] é baseado na contagem de *loops* curtos de ordem l (triângulos para $l = 3$) em redes. Para isso, usa-se o coeficiente de

agrupamento das arestas (i, j), dado por:

$$C_{ij} = \frac{Z_{ij} + 1}{min(k_i - 1, k_j - 1)} \tag{3.2}$$

onde Z_{ij} é o número de triângulos aos quais i e j pertencem e k_i corresponde ao grau do vértice i. Esta medida é baseada no fato que arestas que conectam comunidades tendem a exibir uma valor pequeno para este coeficiente. Este método é rápido ($O(M^4/N^2)$), mas falha se a média do coeficiente de agrupamento da rede é pequena, pois neste caso o valor de C_{ij} é pequeno para todas as arestas.

3.3 Métodos Aglomerativos

Vértices pertencentes a mesma comunidade tendem a apresentar características semelhantes. Baseada nesta premissa é possível obter comunidades considerando a similaridade entre os vértices. Para isso, os métodos aglomerativos iniciam com todos os vértices desconectados e aplicam algum critério de similaridade para, progressivamente, uni-los e obter as comunidades. Estes métodos tem relação direta com teoria de *clustering* [Jain and Dubes 1988], principalmente o ***clustering* hierárquico**, o qual busca descobrir divisões naturais na rede, visto que não se conhece *a priori* quantas comunidades a rede possui.

Para avaliar a similaridade associada com a aresta (i, j) é possível usar várias medidas como, por exemplo, a distância euclidiana e o coeficiente de *Pearson*, apresentadas na subseção 3.3.1. Além destas medidas tem-se a modularidade, que devido à sua importância é apresentada separadamente na subseção 3.3.2.

3.3.1 Medidas de Similaridade

Para avaliar a similaridade associada com a aresta (i, j) uma possibilidade é usar a distância Euclidiana, apresentada na Equação 3.3, ou a correlação de *Pearson* entre vértices, como representada na matriz de adjacência definida pela Equação 3.4.

$$\sqrt{\sum_{k \neq i,j} (a_{ik} - a_{jk})^2} \tag{3.3}$$

$$\frac{\frac{1}{N} \sum_k (a_{ik} - \mu_i)(a_{jk} - \mu_j)}{\sigma_i \sigma_j} \tag{3.4}$$

onde $\mu_i = \frac{1}{N} \sum_j a_{ij}$ e $\sigma_i = \frac{1}{N-1} \sum_j (a_{ij} - \mu_i)^2$. Embora estes métodos sejam rápidos, eles falham com frequência para encontrar as comunidades corretas, onde a estrutura das comu-

nidades já é conhecida. Além disso, eles tendem a encontrar somente os vértices centrais das comunidades (possuem similaridade alta) e deixar de fora os vértices periféricos (possuem menor similaridade) [Newman and Girvan 2004]. Diante disso, uma medida mais usada para identificar comunidades é a modularidade, pois apresenta melhores resultados, mas seu custo computacional é maior.

3.3.2 Medida de Modularidade

O algoritmo divisivo que se baseia na medida de centralidade *Betweenness* para detecção de comunidades, proposto por Girvan e Newman [Girvan and Newman 2002], é um algoritmo eficiente, porém custoso computacionalmente, uma vez que realiza cálculos recursivos para remoção iterativa de arestas que possuem alto grau da medida, apresentando, no pior caso, complexidade de tempo de $O(M^2N)$, para uma rede de M arestas e N vértices (conforme discutido na Subseção 3.2.1).

Em trabalho posterior, Newman [Newman 2004] apresentou uma medida alternativa para detecção de comunidades por meio de um algoritmo (qualificado como *rápido*), cujo custo computacional, no pior caso, possui complexidade de $O((N+M)N)$ e exibe resultados qualitativamente similares aos da medida de *Betweenness* (o que o torna computacionalmente viável quando aplicado a redes de larga escala).

Para tanto, Newman definiu uma função de modularidade Q, que mede a qualidade de uma possível comunidade, ou seja, de uma determinada divisão do grafo ser ou não significativa. Tal função é dada por:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{3.5}$$

onde e_{ii} é a fração das arestas da rede que estão inseridas dentro da comunidade i, e a_i^2 é esta mesma fração, porém considerando que as arestas são inseridas aleatoriamente.

A leitura de tal função é a que segue: valores muito próximos de 0 indicam baixa probabilidade da rede estar dividida em comunidades reais, visto que a chance de tais agrupamentos serem propositais não difere da casualidade de sua formação. Neste sentido, observa-se que quanto mais o resultado for positivo e distante de 0 (valores iguais ou maiores que 0.3 já são considerados significativos), intensifica-se a chance de que tais agrupamentos não existam apenas ao acaso (sua presença está, de alguma forma, intrínseca à estrutura e semântica da rede).

Da forma como foi originalmente definida, Q envolve processos de buscas e divisões iterativas de alto custo computacional, uma vez que deveria-se calcular Q para todas as possíveis

formações de comunidades na rede (o número de termos a serem somados cresce exponencialmente, tornando o cálculo inviável para sistemas com mais de 20 ou 30 vértices). Assim, segundo Newman, a função deve ser mesclada a alguma heurística que permita redução do custo computacional. Apesar de diversas as possibilidades, tais como *Simulated Annealing* [Massen and Doye 2005] e Algoritmos Evolutivos [Tasgin and Bingol 2006], a adotada pelo autor foi um algoritmo de otimização **guloso**.

Partindo-se de um estado no qual cada vértice da rede representa uma comunidade, comunidades são conectadas duas a duas, repetidamente, até que seja selecionada a conexão que resultar no maior valor de Q. O algoritmo (aglomerativo) segue até que toda a rede seja considerada uma única comunidade. A variação em Q após a conexão entre duas comunidades i e j pode ser medida como segue:

$$\Delta Q = 2(e_{ij} - a_i a_j) \tag{3.6}$$

onde e_{ij} é a fração das arestas que conectam a comunidade i à comunidade j, a_i é fração total de arestas que conectam a comunidade i às demais comunidades da rede e pode ser calculada por $a_i = \sum_k e_{ik}$, assim como a_j é a fração total de arestas que conectam a comunidade j às demais comunidades da rede e pode ser calculada da mesma forma que a_i. Todo processo é representado na forma de um *dendrograma* (árvore que exibe a ordem das conexões), conforme mostra a Figura 3.2, que apresenta o resultado final e as possíveis comunidades encontradas na rede do clube de karate de Zachary [Zachary 1977].

Apesar do algoritmo de Newman ser largamente citado como referência em diversas pesquisas na área, ele também sofre críticas, as quais alegam, principalmente, que sua função de modularidade é uma medida imprecisa, uma vez que se baseia na diferença entre o proposital e o aleatório. Assim sendo, vários trabalhos significativos emergiram inspirados na novidade computacional do conceito de modularidade, porém propondo melhorias em sua definição e aplicação. Dentre estes trabalhos têm-se a otimização extrema e a otimização por meio do método de Monte Carlo com *Simulated Annealing* e *Basin Hopping*, os quais serão apresentados com mais detalhes na Seção 3.6.

3.4 Métodos Espectrais

Os métodos espectrais são baseados na análise dos autovetores das matrizes derivadas das redes complexas. A quantidade medida corresponde aos autovalores das matrizes associadas com a matriz de adjacência A. Estas matrizes podem ser a matriz Laplaciana ($L = D - A$) ou a matriz Normal ($\tilde{A} = D^{-1}A$), onde D é a matriz diagonal dos graus dos vértices com elementos

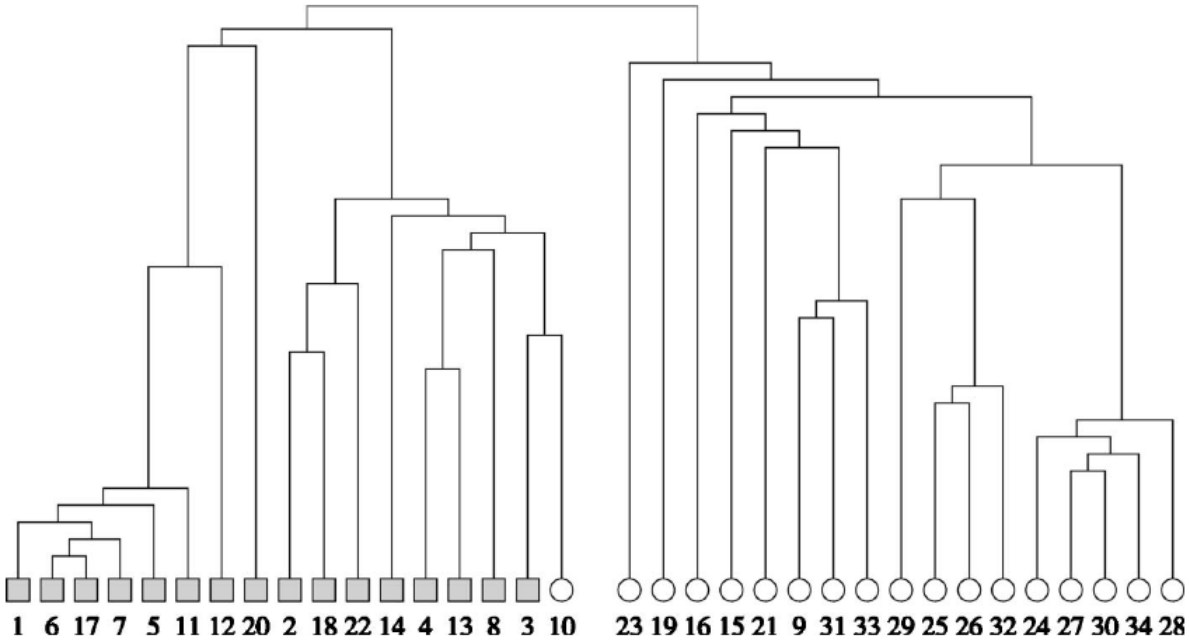

Figura 3.2: Dendograma que apresenta duas comunidades encontradas (através do formato e cores dos vértices) na rede do clube de karate de Zachary [Zachary 1977].

$d_{ii} = \sum_j a_{ij}$, $d_{ij} = 0$ para $i \neq j$ [Seary and Richards 1995].

Alguns métodos espectrais foram discutidos em Newman [Newman 2006], o qual também propôs um método que reformula o conceito de modularidade em termos dos autovetores de uma nova matriz que caracteriza a rede, chamada de matriz de modularidade. Para cada subgrafo g, sua matriz de modularidade $B^{(g)}$, para os vértices i e j pertencentes a g, tem seus elementos dados por:

$$b_{ij}^{(g)} = a_{ij} - \frac{k_i k_j}{2M} - \delta_{ij} \sum_{u \in g} [a_{iu} - \frac{k_i k_u}{2M}] \tag{3.7}$$

onde a_{ij} corresponde ao número de arestas entre i e j (as medidas a_{ij} são os elementos da matriz de adjacência), $\frac{k_i k_j}{2M}$ é o número esperado de arestas ente os vértices i e j caso tenham sido colocadas aleatoriamente (k_i e k_j representam o grau dos vértices i e j, respectivamente) e $\delta_{ij} = \mathrm{v}_i^T \mathrm{v}_j$, onde v representa o autovetor.

Para dividir a rede em comunidades, considerando a matriz de modularidade $B^{(g)}$, deve-se encontrar seu autovalor mais positivo com seu correspondente autovetor. De acordo com os sinais dos elementos do autovetor, a rede é dividida em duas partes: vértices com elementos positivos são atribuídos a uma comunidade e vértices com elementos negativos são atribuídos a outra. Em seguida, o processo é repetido recursivamente para cada comunidade até obter

uma divisão da rede onde zero ou uma contribuição negativa para a modularidade total seja encontrada [Newman 2006].

3.5 Métodos Locais

Em algumas redes complexas a maioria das comunidades são conectadas a somente uma fração das comunidades remanescentes, ou seja, as comunidades são mais conectadas localmente entre si. Por isso, é necessário o desenvolvimento de métodos que levam em conta a conectividade local da comunidade e superam a dependência global da rede. Alguns métodos foram propostos com base no cálculo da medida de **modularidade local**, a qual faz uma avaliação da comunidade atual e de suas comunidades vizinhas.

Clauset [Clauset 2005] propôs um método cuja ideia principal é o crescimento passo-a-passo da comunidade junto com a exploração da rede. Dessa forma, a comunidade C inicia-se com somente um vértice v_0 e quando este vértice é explorado, a lista de seus vizinhos é conhecida. Definindo o conjunto U pela lista de todos os vértices que não estão em C, mas são adjacentes a alguns de seus vértices e, o conjunto B pelo subconjunto dos vértices de C que são adjacentes a pelo menos um vértice em U, tem-se a modularidade local dada pela razão entre o número de arestas com um vértice final em B e nenhum vértice final em U e, o número de arestas com vértices finais em B, conforme apresentado na equação 3.8, considerando redes não direcionadas.

$$R = \frac{\sum_{i \in B, j \in C} a_{ij}}{\sum_{i \in B, j} a_{ij}} \tag{3.8}$$

O algoritmo consiste em escolher, iterativamente do conjunto U, o vértice que resultará no maior aumento (ou menor diminuição) do valor de R quando adicionado à comunidade C. A iteração termina quando um número pré-definido de vértices for incluído na comunidade C.

Já Muff [Muff et al. 2005] propôs uma modificação na medida de modularidade global, transformando-a em uma medida de modularidade local. Com isso, a contribuição da modularidade para cada comunidade i é calculada para uma subrede, que consiste da comunidade i e suas comunidades vizinhas. Isso requer a determinação da vizinhança de i ou, mais precisamente, de todas as arestas que estão contidas nesta vizinhança. A soma de todas as contribuições para as K comunidades resulta na medida de modularidade local, dada pela equação 3.9:

$$LQ = \sum_{i=1}^{K} [\frac{L_i}{L_{iC}} - \frac{(L_i)_{in}(L_i)_{out}}{(L_{iC})^2}] \tag{3.9}$$

onde N é a quantidade de arestas da rede, Δ_i é o conjunto de comunidades vizinhas de i, L_i é o total de arestas dentro da comunidade i, $(L_i)_{in}$ é o total de arestas que chegam na comunidade i, $(L_i)_{out}$ é o total de arestas que saem da comunidade i e L_{iC} é o total de arestas internas em cada comunidade $C \in \Delta_i$, onde Δ_i é o conjunto das comunidades vizinhas de i. Quanto mais comunidades localmente conectadas a rede possuir, maior será o valor de LQ. Diferentemente da medida de modularidade Q, LQ não é limitada por 1, mas pode levar à qualquer valor.

Se todas as comunidades da rede forem conectadas entre si, a medida LQ coincidirá com Q. Ressalta-se que ambas as medidas podem ser usadas como uma função de *fitness* em um processo de otimização para detecção de comunidades em uma rede. Em seu artigo, Muff [Muff et al. 2005] comparou a otimização de Q e LQ em redes biológicas e conclui que as duas medidas fornecem uma visualização em diferentes profundidades da estrutura de comunidades da rede. Q provê uma visualização global, enquanto LQ considera cada comunidade individualmente e suas comunidades vizinhas, produzindo comunidades com alta confiança, desprezando o restante da rede. Além disso, observou-se que LQ permite a obtenção de um número muito maior de comunidades que Q. No entanto, acredita-se que ambas as medidas produzem informações complementares.

3.6 Maximização da Modularidade

Conforme visto na Subseção 3.3.2, a medida Q de Newman, exibida pela Equação 3.5, apresenta uma forma alternativa (ou seja, não mais baseando-se em distâncias ou outros conceitos de similaridade) para se calcular o quão relevante é um agrupamento de vértices em uma rede complexa, ou seja, estima a qualidade do particionamento da rede em comunidades. Esta medida deu origem a muitos outros trabalhos que buscam seu refinamento, a fim de se obter como resposta, tanto melhor precisão na detecção de comunidades, quanto maior desempenho e redução do custo computacional. Algumas destas pesquisas serão descritas nas subseções seguintes.

3.6.1 Otimização Extrema

Como a busca pelo valor da modularidade ótimo é um problema NP-Difícil, uma estratégia de busca heurística é obrigatória para restringir o espaço de busca enquanto preserva o objetivo da otimização. Diante disso, Duch e Arenas [Duch and Arenas 2005] propuseram um algoritmo de divisão que otimiza a modularidade Q por meio de uma busca baseada no algoritmo de Otimização Extrema (EO), desenvolvido por Boettcher [Boettcher 2001]. Tal algoritmo funciona por meio da otimização de uma variável global (modularidade) através do melhoramento de variáveis locais. Estas variáveis locais devem estar relacionadas com a contribuição do vértice individual i para o somatório de equação Q, dada uma certa divisão em c comunidades:

$$q_i = K_{c(i)} - k_i a_{c(i)} \tag{3.10}$$

onde $K_{c(i)}$ corresponde ao número de arestas, que o vértice i pertencente à comunidade c, tem com vértices na mesma comunidade, k_i é o grau do vértice i e $a_{c(i)}$ é a fração das arestas que possui pelo menos o vértice i dentro da comunidade c. Com isso, tem-se que:

$$Q = \frac{1}{2M} \sum_i q_i \tag{3.11}$$

onde i corresponde a todos os vértices da rede dada certa divisão em comunidades e, M é o número total de arestas da rede. A equação 3.10 provê uma medida que depende do grau do vértice e sua normalização envolve todas as arestas da rede depois do somatório. Redimensionado a variável local q_i através do grau do vértice i obtém-se a definição adequada da contribuição do vértice i para a modularidade, em relação ao seu próprio grau e normalizada no intervalo $[-1, 1]$.

$$\lambda_i = \frac{q_i}{k_i} = \frac{K_{c(i)}}{k_i} - a_{c(i)} \tag{3.12}$$

A variável λ_i é chamada de *fitness* do vértice i e corresponde à variável local envolvida no processo de otimização extrema. Baseado no *fitness*, o processo de busca heurística para encontrar o valor de modularidade ótimo consiste em, inicialmente, dividir aleatoriamente a rede em duas partições com o mesmo número de vértices (os componentes conectados em cada partição serão entendidos como comunidades). A cada passo o sistema se auto-organiza, movimentando o vértice com menor *fitness* (extremo) de uma partição para outra. O processo é repetido até que o valor máximo de Q seja obtido. Depois disso, deletam-se todas as arestas entre ambas as partições e procede-se, recursivamente, com todos os componentes conectados resultantes. O processo termina quando a modularidade Q não puder ser melhorada.

Para ilustrar o processo, considere a rede do clube de karate de Zhacary [Zachary 1977]. Inicialmente os vértices foram divididos em duas partições aleatórias (Figura 3.3 - esquerda). O número de comunidades iniciais (componentes conectados em cada partição) neste caso é cinco (Figura 3.3 - direita). Em seguida, o vértice com menor *fitness* é selecionado e movido de uma partição para outra e, este movimento provoca uma avalanche de modificações no *fitness* do restante da rede. Calcula-se o novo valor da modularidade e repete-se o processo até que não haja mais mudanças em seu valor. Nesta rede, foram necessárias três iterações recursivas, conforme apresentado na Figura 3.4.

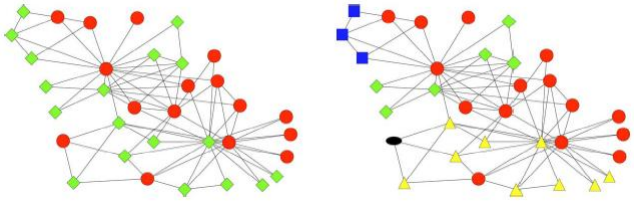

Figura 3.3: Esquerda: Divisão da rede de Zachary em duas partições. Direita: Cinco comunidades identificadas como componentes conectados em cada partição. Figura adaptada de [Duch and Arenas 2005]

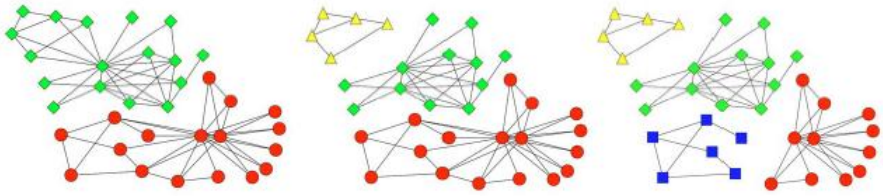

Figura 3.4: Rede após a remoção das arestas em cada processo de corte. Figura adaptada de [Duch and Arenas 2005]

Ressalta-se que no algoritmo de otimização extrema original [Boettcher 2001], o vértice selecionado é aquele com pior valor de λ_i. Este procedimento é dependente da divisão inicial da rede e não há possibilidade de escapar do máximo local. Diante disse, o algoritmo de Duch e Arenas [Duch and Arenas 2005] optou por utilizar a seleção probabilística chamada $\tau - EO$, na qual os vértices são ordenados de acordo com seus valores de *fitness*, e então o vértice i do *rank* é selecionado de acordo com a seguinte distribuição de probabilidade:

$$P(i) \propto i^{-\tau} \tag{3.13}$$

Esta seleção é menos sensível a diferentes inicializações e permite escapar do máximo local. O expoente $-\tau$ tem sido modulado em torno dos valores ótimos obtidos de redes aleatórias de

tamanho N que aproximam a escala $\tau \sim 1 + \frac{1}{ln(N)}$. O uso desta técnica também implica na determinação do número de passos de auto-organização αN necessários para decidir que o valor máximo de Q tem poucas chances de ser melhorado. Na prática, em cada passo, verifica-se o último valor máximo obtido para Q; se este não é melhorado em αN passos, termina-se a busca. Geralmente usa-se $\alpha = 1$, permitindo tantos passos quanto o número de vértices, para melhorar o valor máximo do Q atual.

O custo computacional de todo o processo é $O(N^2 ln^2 N)$, onde $NlnN$ é o custo do processo de ordenação, que pode ser reduzido para $O(N)$ usando a estrutura de dados *heap*. Dessa forma, o custo total do algoritmo passa para $O(N^2 lnN)$. Não é um algoritmo rápido, mas, os resultados obtidos apresentaram altos valores de modularidade e precisão na determinação da estrutura das comunidades, quando comparados com a otimização da modularidade proposta por Newman[Newman 2004].

3.6.2 Otimização usando o método de Monte Carlo com *Simulated Annealing* e *Basin Hopping*

Massen e Doye [Massen and Doye 2005] propuseram a identificação de comunidades em redes de superfícies de energia potencial [Dorogovtsev and Mendes 2003] por meio da otimização da modularidade Q usando o método Monte Carlo com *Simulated Annealing* e *Basin Hopping*. Ambos são usados frequentemente em problemas de otimização. Na aproximação baseada no método de Monte Carlo com *Simulated Annealing*, a cada passo, um vértice e uma comunidade são escolhidos aleatoriamente. A comunidade pode ser qualquer uma das comunidades existentes, inclusive aquela que contém o vértice selecionado, ou uma nova comunidade que não contém nenhum vértice.

Movendo o vértice de sua comunidade inicial para uma nova comunidade modifica-se Q para ΔQ. Se $\Delta Q > 0$, o movimento é aceito, senão, ele é aceito com probabilidade $exp(\beta \Delta Q)$. Este é o critério Metropolis, onde β representa a temperatura inversa. Em altas temperaturas muitos movimentos são aceitos e muitas divisões diferentes de comunidades são experimentadas, enquanto que em baixas temperaturas, poucas divisões são experimentadas, mas, estas geralmente apresentam altos valores de modularidade.

O princípio do *Simulated Annealing* envolve iniciar o algoritmo com altas temperaturas e diminuir a temperatura até Q se tornar constante, pois nenhum outro movimento será aceito. Espera-se que, com o uso do *Simulated Annealing*, o particionamento final seja o ótimo global, no entanto também pode resultar em um ótimo local. Para aumentar a probabilidade de sucesso, *quenches* podem ser aplicados periodicamente. Nestes *quenches*, ΔQ é calculado movendo

todos os vértices da comunidade selecionada para todas as outras comunidades e o movimento com maior ΔQ é aceito. Este processo é repetido até o maior ΔQ ser menor ou igual a zero, implicando que o valor de Q não poderá ser melhorado.

O *Simulated Annealing* executa rapidamente e encontra altos valores de modularidade, no entanto, não é um método de otimização global eficiente. Diante disso, a aproximação *Basin Hopping* é utilizada, visto que obtém altos valores de modularidade, executa bem em várias aplicações [Clauset et al. 2004] e também utiliza o método de Monte Carlo. Primeiro, cada passo consiste da mudança aleatória de uma série de vértices das comunidades, não apenas um. Segundo, depois de cada passo, aplica-se o processo de *quenche* à nova partição e os valores de modularidade das partições após o *quenche* são submetidos ao critério de aceitação de Metropolis. Então, se o passo é aceito, a partição corrente é atualizada para a partição resultante do processo de *quenche*. Este algoritmo é mais lento para terminar sua execução, no entanto obtém altos valores de modularidade rapidamente.

Em ambas as aproximações (*Simulated Annealing* e *Basin Hopping*) o ponto inicial pode ser qualquer partição de vértices em comunidades, inclusive N comunidades de um nó. No entanto, estes algoritmos são mais rápidos se a partição inicial é obtida de um algoritmo guloso ou de um particionamento em comunidades obtido de maneira aglomerativa similar, mas, usando o método de Monte Carlo, o qual é mais rápido. Para criar esta partição inicial, de preferência checando todas as possíveis arestas como em um algoritmo guloso, uma única aresta é escolhida aleatoriamente e inserida se ΔQ satisfaz o critério de Metropolis.

O *Simulated Annealing* executa melhor quando é iniciado com uma partição obtida de um algoritmo guloso, a qual obteve altos valores de Q. Caso este algoritmo seja iniciado com uma partição aleatória, os resultados dependerão fortemente de parâmetros inicias como, taxa de resfriamento e quantidade de *quenches*. Já no algoritmo *Basin Hopping*, as condições iniciais tem pouco efeito no Q_{max}, então ele pode ser iniciado com partições aleatórias, tornado-se mais adequado para grandes redes, visto que o algoritmo guloso, embora rápido, requer grandes vetores para grandes redes.

Capítulo 4

Aplicações e Desafios na Área

Muitas redes reais apresentam em sua estrutura diversas comunidades, cuja identificação é muito importante dado que vértices pertencentes a mesma comunidade tem maiores chances de compartilharem propriedades e dinâmicas. Além disso, a quantidade e as características das comunidades provê subsídios para determinar o tipo da rede, assim como entender a sua organização.

A identificação de comunidades vem sendo aplicada em diversas redes reais, dentre elas cita-se as redes sociais, a Internet, a *World Wide Web (WWW)*, as redes biológicas, as redes organizacionais e de negócios, as redes de *e-mail* e de chamadas telefônicas, entre outras. As redes sociais foram uma das primeiras redes modeladas, pois sabia-se que as pessoas, geralmente, se dividem em grupos de interesse, ocupação, idade etc. Com isso, a detecção de comunidades nestas redes foi utilizada para observar o comportamento da sociedade e as relações entre indivíduos.

Já a Internet é uma das maiores redes reais existentes e seu estudo fornece informações sobre as comunidades formadas por roteadores geograficamente próximos um dos outros, as quais podem ser consideradas com o objetivo de melhorar o fluxo de dados. No caso da *WWW*, páginas relacionadas ao mesmo assunto são tipicamente organizadas em comunidades, então a identificação destas comunidades pode ajudar na tarefa de procura por informação.

Dentre as redes biológicas pode-se citar as redes neurais, as redes de reações metabólicas e de interações proteína-proteína. Todas tem atraído bastante atenção, pois permitem um maior entendimento dos sistemas da vida. Nas redes neurais, onde vértices representam neurônios e arestas representam as conexões, geralmente estuda-se seus aspectos estruturais. Já nas redes de reações metabólicas, os processos químicos podem ser representados por grafos de reações químicas. Por fim, nas redes de interação proteína-proteína cada vértice é um proteína e as arestas representam interações físicas entre proteínas. Ressalta-se que es-

tas redes são adequadas para a aplicação da modularidade global e local [Muff et al. 2005]. Detalhes das redes citadas e diversas outras redes reais podem ser encontrados Dorogovtesev [Dorogovtsev and Mendes 2003].

Nota-se que grande parte das redes reais são de larga-escala e as diferentes comunidades destas redes podem ter propriedades distintas que são perdidas quando é feita a analisa de toda a rede. Por isso, é muito importante a aplicação de algoritmos de detecção de comunidades. No entanto, sabe-se que o problema de particionamento da rede em comunidades é um problema NP-Difícil, sendo necessária a utilização de vários métodos heurísticos que buscam minimizar o custo computacional sem perder a qualidade das comunidades encontradas.

Percebe-se que os principais desafios da área de detecção de comunidades é o custo computacional apresentado pelos algoritmos utilizados atualmente. Além disso, o número e o tamanho das comunidades não é conhecido a *priori*, logo eles devem ser estabelecidos pelos algoritmos de detecção. A grande maioria dos trabalhos ainda se baseiam na medida de modularidade para estimar a qualidade da estrutura de comunidades obtida, no entanto, tal medida é tida como imprecisa por alguns pesquisadores devido ao fato de se basear na diferença entre o proposital e o aleatório. Juntando-se a isso tem-se o custo computacional apresentado pelos algoritmos de otimização usados para obter melhores valores da modularidade.

Capítulo 5

Conclusões

Este trabalho apresentou uma revisão bibliográfica sobre Detecção de Comunidades em Redes Complexas. Foram abordados conceitos introdutórios sobre a estrutura de comunidades e algoritmos desenvolvidos para sua detecção, enfatizando a relevância do tema na área de aprendizado de máquina.

O trabalho de Newman [Newman and Girvan 2004] sobre a medida de Modularidade é um dos principais artigos citados na literatura, dando margens à pesquisas posteriores que visam a otimização e maximização da medida. E, dado que a descoberta de comunidades é um problema NP-Difícil, vários destes trabalhos buscam embutir uma heurística mais adequada, de acordo com suas propostas de modificações matemáticas.

Uma vez que Redes Complexas podem modelar sistemas reais, a detecção de determinados agrupamentos nestes sistemas pode levar a descoberta de características específicas e à análise da organização e evolução dinâmica dos mesmos. Trata-se, portanto, de uma área de pesquisa em evidência e ascensão que requer o desenvolvimento de algoritmos que busquem maior precisão em suas detecções e, ao mesmo tempo, que proporcionem um maior desempenho, reduzindo o custo computacional para a execução de tal tarefa.

Referências Bibliográficas

[Boettcher 2001] Boettcher, S. (2001). Extremal Optimization for Graph Partitioning. *Physical Review E*, 64(026114).

[Clauset 2005] Clauset, A. (2005). Finding Local Community Structure in Networks. *Physical Review E*, 72(026132).

[Clauset et al. 2004] Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding Community Structure in Very Large Networks. *Physical Review E*, 79(066111).

[Costa et al. 2007] Costa, L. F., Rodrigues, F. A., Villas Boas, P. R., and Travieso, G. (2007). Characterization of Complex Networks: A Survey of Measurements. *Advances in Physics*, 56:167–242.

[Dorogovtsev and Mendes 2003] Dorogovtsev, S. N. and Mendes, J. F. F. (2003). *Evolution of Networks - From Biological Nets to the Internet and WWW*. Oxford University Press Inc, New York.

[Duch and Arenas 2005] Duch, J. and Arenas, A. (2005). Community Detection Complex Networks using Extremal Optimization. *Physical Review E*, 72.

[Ester et al. 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231.

[Girvan and Newman 2002] Girvan, M. and Newman, M. E. J. (2002). Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Science USA*, 99(12):7821–7826.

[Jain and Dubes 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

[Karypis et al. 1999] Karypis, G., Han, E. H., and Kumar, V. (1999). CHAMELEON: A Hhierarchical Clustering Algorithm using Dynamic Modeling. *Computer*, 32(8):68–75.

[Massen and Doye 2005] Massen, C. P. and Doye, J. P. K. (2005). Identifying Communities within Energy Landscapes. *Physical Review E*, 71(046101).

[Muff et al. 2005] Muff, S., Rao, F., and Caflisch, A. (2005). Local Modularity Measure for Network Clusterizations. *Physical Review E*, 72(056107).

[Newman 2004] Newman, M. E. J. (2004). Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69(066133).

[Newman 2006] Newman, M. E. J. (2006). Finding Community Structure in Networks using the Eigenvectors of Matrices. *Physics/0605087*.

[Newman and Girvan 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and Evaluating Community Structure in Networks. *Physical Review E*, 69(026113).

[Oliveira et al. 2008] Oliveira, T. B. S., Zhao, L., Faceli, K., and Carvalho, A. C. P. L. (2008). Data Clustering Based on Complex Network Community Detection. *Proceedings of 2008 IEEE World Congress on Computational Intelligence (WCCI 2008) - IEEE Computer Society*, 1:2121–2126.

[Quiles et al. 2008] Quiles, M. G., Zhao, L., Alonso, R. L., and Romero, R. A. F. (2008). Particle Competition for Complex Network Community Detection. *Chaos (Woodbury)*, 18(033107):1–10.

[Radicchi et al. 2004] Radicchi, C., Castellano, F., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and Identifying Communities in Networks. *Proceedings of the National Academy of Science USA*, 101(9):2658–2663.

[Reichadt and Bornholdt 2006] Reichadt, J. and Bornholdt, S. (2006). Statistical Mechanics of Community Detection. *cond-mat/0603718*.

[Seary and Richards 1995] Seary, A. J. and Richards, W. D. (1995). Partitioning networks by eigenvectors. *Proceedings of the International Conference on Social Networks*, 1.

[Stauffer et al. 2003] Stauffer, D., Aharony, A., C. L. F., and Adler, J. (2003). Efficient Hopfield Pattern Recognition on a Scale-Free Neural Network. *The European Physical Journal B*, 32:395–399.

[Tasgin and Bingol 2006] Tasgin, M. and Bingol, H. (2006). Community Detection in Complex Networks using Genetic Algorithms. *cond-mat/0604419*.

[Zachary 1977] Zachary, W. W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Anthropological Research*, 33:452–473.