Comparação de Algoritmos de Detecção de Comunidades em Redes Complexas

Fabiano Berardo de Souza, Glenda Michele Botelho
Universidade de São Paulo (USP)
Instituto de Ciências Matemáticas e de Computação
São Carlos - SP - Brasil
fabber.usp@gmail.com, glenda@icmc.usp.br

Resumo

Existem vários algoritmos de detecção de comunidades em redes complexas na literatura. No entanto, novas pesquisas tem surgido com o objetivo de detectar comunidades de forma eficiente e com menor custo computacional. Este trabalho apresenta uma comparação entre cinco algoritmos de detecção de comunidades (Betweenness [7], Caminhada Aleatória [15], FastGreedy [4], Baseado em Autovalores e Autovetores [13] e Spinglass [18]) por meio da medida de Modularidade [11], tempo de execução e número de comunidades encontradas. As partições obtidas pelos diferentes algoritmos apresentaram valores altos de modularidade e percebeu-se a influência da quantidade de vértices e arestas no tempo de execução de alguns dos algoritmos.

Palavras-chave: redes complexas, detecção de comunidades, modularidade.

1. Introdução

Detecção de comunidades em redes complexas é uma área de pesquisa recente e em ascensão que envolve técnicas de *Clustering* presentes em aprendizado de máquina, mais especificamente, no modelo de aprendizado não-supervisionado. Porém, muitos dos algoritmos clássicos para detecção de agrupamentos em grafos tornam-se insuficientes quando aplicados à redes mais complexas, com um número muito grande de vértices e arestas e que modelam sistemas reais, possuindo, assim, um comportamento dinâmico.

Muitas propostas de algoritmos para esta finalidade vêm surgindo na literatura, a maioria embasada no modelo hierárquico de *Clustering*, visto que não se conhece *a priori* o número e o tamanho das comunidades presentes na rede e este modelo permite a geração de uma árvore (dendograma) que exibe a ordem de formação dos agrupamentos para diferentes quantidades de grupos. Dentre estes algoritmos, pode-se citar os trabalhos de [19], [7], [12], [1], [14], [4], [11], [13], [3], [10], [5], [9], [2], [20], [16], entre outros.

Uma vez que a definição de comunidade não é algo claro e que muitas vezes depende do contexto do sistema que se está analisando, avaliar um algoritmo que detecta comunidades torna-se, da mesma forma, algo não trivial. Porém, Newman [11] desenvolveu uma função de modularidade que mede o quão significante é uma certa divisão da rede. Portanto, com o objetivo de comparar o desempenho de diversos algoritmos para detecção de comunidades em redes complexas, este trabalho utiliza a função de modularidade, além de tempo de execução e número de comunidades obtidas.

A seção 2 destaca a medida de modularidade desenvolvida por Newman [11] para estimar a qualidade das comunidades encontradas em uma rede. A seção 3 apresenta informações referentes aos experimentos utilizados, destacando o ambiente de programação, as bases de dados, os algoritmos comparados e a versão da medida de modularidade implementada. A seção 4 apresenta os resultados obtidos. Por fim, a seção 5 aborda as conclusões.

2. Medida de Modularidade

Em trabalho publicado em 2004, Newman [11] definiu uma função de modularidade Q, que mede a qualidade de uma possível divisão da rede em comunidades, ou seja, de uma determinada divisão do grafo ser ou não significativa. Tal função é dada por:

$$Q = \sum_{i} (e_{ii} - a_i^2) \tag{1}$$

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).

No mesmo trabalho, Newman propõe o uso desta medida juntamente com um algoritmo hierárquico aglomerativo guloso, no qual partindo-se de um estado em que cada vértice representa uma comunidade, comunidades são conectadas duas a duas, repetidamente, até que todos os vértices façam parte da mesma comunidade. Q é calculada para o estado inicial e a cada fusão entre duas comunidades i e j, o valor da variação em Q pode ser medido como segue:

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

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. Desta forma, a divisão da rede que obtiver o máximo resultado de Q será considerada a melhor divisão possível da rede em comunidades.

3. Experimentos

Com a intenção de comparar o desempenho de alguns algoritmos de detecção de comunidades em redes complexas, este trabalho utiliza a medida de modularidade de Newman [11] para extrair o quão significativa foi a divisão da rede em comunidades que o algoritmo encontrou. Além desta foram utilizadas duas outras medidas: tempo de execução do algoritmo e o número de comunidades encontradas (no caso da melhor medida de modularidade).

Para permitir a realização dos experimentos diversos recursos foram utilizados e são especificados nas subseções seguintes: o ambiente de programação (subseção 3.1), as bases de dados (subseção 3.2), os algoritmos de detecção (subseção 3.3) e a versão da medida de modularidade implementada na ferramenta usada (subseção 3.4).

3.1. Ambiente

Os experimentos foram executados em uma máquina com processador Core Duo de 1.73GHz e 2GB de memória RAM contendo sistema operacional Linux (Slackware, versão 13.1) de 32 bits. Utilizou-se o ambiente estatístico R [17] como ambiente de programação e o pacote de bibliotecas IGRAPH [8], o qual implementa muitos dos algoritmos aqui citados para detecção de comunidades.

3.2. Bases Utilizadas

Quatro redes complexas foram selecionadas visando mesclar diferentes tamanhos, complexidade e outras características, conforme visto a seguir:

- *Zachary's Karate Club* [22]: rede social de relacionamentos entre 34 membros de um clube de karate de uma Universidade dos Estados Unidos. A rede é representada por um grafo com 34 vértices e 78 arestas não-direcionadas.
- *American College Football* [6]: rede de jogos de futebol americano entre colégios, da primeira divisão. A rede é representada por um grafo com 115 vértices (cada um representando um time) e 615 arestas (jogos entre dois times) não-direcionadas.
- *Neural Network* [21]: rede direcionada e ponderada representando a rede neural de C. Elegans. No total tem-se um grafo com 297 vértices e 2359 arestas.
- *Coauthorship Network Science (Netscience)* [12]: rede de co-autoria de cientistas. No total tem-se um grafo desconexo com 1589 vértices e 2742 arestas não direcionadas e ponderadas.

Tentou-se, também, utilizar a rede *Internet*, a qual contém 22.962 vértices. Mas, conforme será observado mais adiante, os recursos computacionais foram insuficientes para tratar esta base. Todas estas redes estão disponíveis no site do pesquisador Mark Newman, cujo endereço é *http://www-personal.umich.edu/ mejn/*.

3.3. Algoritmos de Detecção Utilizados

Todos os algoritmos escolhidos estão presentes na biblioteca IGRAPH [8]. Visou-se a seleção de algoritmos pertencentes a diferentes abordagens (divisivos, aglomerativos, espectrais e otimização da modularidade) com o intuito de comparar os respectivos desempenhos. Os algoritmos selecionados foram:

- *Betwenness* [7]: algoritmo divisivo que se baseia na contagem de caminhos mínimos entre vértices. Dessa forma, define-se que arestas com valor baixo de *Betwenness* pertencem a mesma comunidade e arestas com alto valor de *Betwenness* separam diferentes comunidades.
- *Caminhada Aleatória* [15]: algoritmo que considera que caminhos aleatórios curtos tendem a estar na mesma comunidade.

- *FastGreedy* [4]: algoritmo que otimiza a medida de modularidade original de Newman [11] utilizando uma busca gulosa.
- *Autovetores* [13]: algoritmo que reformula o conceito de modularidade em termos dos autovetores e autovalores de uma nova matriz, a matriz de modularidade.
- *Spinglass* [18]: algoritmo que otimiza a modularidade de Newman usando *Simulated Annealing*.

Ressalta-se que detalhes dos algoritmos estão no primeiro trabalho.

3.4. Medida de Modularidade Utilizada

A função de modularidade que a biblioteca IGRAPH implementa é uma versão otimizada da medida original e foi proposta por Newman e Girvan em [14]. A equação seguinte apresenta a modularidade Q:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \delta(c_i cj) \right] \quad (3)$$

onde m é o número de arestas da rede, A_{ij} é o elemento ij da matriz de incidência A, k_i é o grau do vértice i, k_j é o grau do vértice j, c_i é o valor do componente do vértice i, c_j é o mesmo para o vértice j. A função δ retorna 1 caso i e j pertençam ao mesmo componente e 0 caso contrário. O somatório percorre todos os pares de vértices i j da rede.

Esta função na IGRAPH recebe a rede original, a árvore (dendograma) que contém a hierarquia da formação das comunidades e um parâmetro que indica qual a posição na árvore que se deseja analisar. Diante disso, nossa proposta, tal qual aconselha Newman em seu trabalho original, foi a de promover um laço para varrer toda árvore e selecionar a divisão que retornar o maior valor de modularidade. Porém, diferentemente de Newman, não calculamos a variação em Q, e sim, calculamos diretamente Q para cada nível da árvore.

4. Resultados

Os algoritmos de detecção de comunidades escolhidos foram avaliados considerando três medidas: modularidade, tempo de execução (medido em segundos - s) e número de comunidades obtidas para o maior valor de modularidade. A Tabela 1 apresenta os resultados referentes à rede do clube de karate de Zachary [22] e a Figura 1 compara graficamente as medidas de modularidade obtidas pelos diferentes algoritmos. Percebe-se que todos os algoritmos obtiveram valores de modularidade satisfatórios, baixo tempo de execução e número de comunidades semelhante.

A Tabela 2 apresenta os resultados obtidos na rede *Footboll* pelos diferentes algoritmos de detecção e a Figura 2 demonstra graficamente as medidas de modularidade encontradas. Ressalta-se que o algoritmo *FastGreedy* não conseguiu trabalhar com esta rede. Já os outros algoritmos obtiveram valores altos de modularidade, quantidades de comunidades semelhantes e tempo de execução baixo (principalmente o algoritmo *Caminhada Aleatória*).

	Modularidade	Tempo(s)	N° de Comunidades
Betwenness	0,4012985	0,019	5
Caminhada Aleatória	0,3532216	0,001	5
FastGreedy	0,3806706	0,001	3
Autovetores	0,3776298	0,031	5
Spinglass	0,4188034	2,118	4

Tabela 1. Rede do clube de *Karate*.

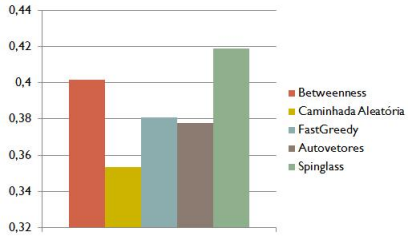

Figura 1. Modularidade obtida na rede do clube de *Karate* pelos diferentes algoritmos de detecção.

	Modularidade	Tempo(s)	N° de Comunidades
Betwenness	0,6005129	2,394	10
Caminhada Aleatória	0,6038112	0,007	10
FastGreedy	*	*	*
Autovetores	0,4402313	0,120	13
Spinglass	0,6027933	3,117	10

Tabela 2. Rede *Football*.

Os resultados contidos na Tabela 3 são referentes a rede *Neural Network* e a Figura 3 apresenta a modularidade obtida para esta rede considerando os diferentes algoritmos. Ressalta-se que os algoritmos *FastGreedy* e *Autovetores* não trabalham com grafos direcionados. Já os resultados dos outros algoritmos foram bem diferentes. O algoritmo *Betwenness* apresentou baixo valor de modularidade e um alto número de comunidades. O algoritmo *Caminhada Aleatória* foi o que apresentou maior valor de modularidade e menor tempo de execução. Por fim, o algoritmo *Spinglass* apresentou valor de modularidade satisfatório e o maior tempo de execução.

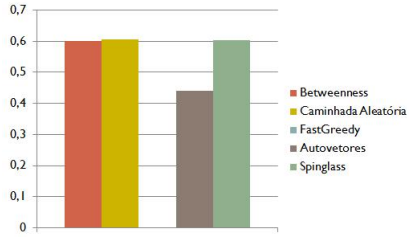

Figura 2. Modularidade obtida na rede *Football* pelos diferentes algoritmos de detecção.

	Modularidade	Tempo(s)	N° de Comunidades
Betwenness	0,0887715	22,493	200
Caminhada Aleatória	0,469383	0,045	24
FastGreedy	*	*	*
Autovetores	*	*	*
Spinglass	0,3883311	30,552	5

Tabela 3. Rede *Neural Network*.

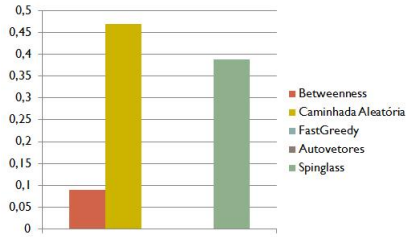

Figura 3. Modularidade obtida na rede *Neural Network* pelos diferentes algoritmos de detecção.

Por fim, a Tabela 4 apresenta os resultados para a rede *Netscience* e a Figura 4 mostra graficamente os valores de modularidade obtidos pelos diferentes algoritmos para esta rede. Ressalta-se que os algoritmos *Caminhada Aleatória* e *Spinglass* não trabalham com grafos desconexos. Já os outros algoritmos apresentaram valores de modularidade elevados (provavelmente devido ao grafo ser desconexo) e resultaram em um número diferente de comunidades. O algoritmo *FastGreedy* executou rapidamente, mesmo com elevado número de vértices presente nesta rede.

Para permitir um melhor entendimento dos resultados, as medidas foram analisadas em conjunto por meio de diferentes gráficos. As Figuras 5 e 6 apresentam, respectivamente, a relação entre medida de modularidade versus quantidade de vértices e medida de modularidade versus quantidade de arestas. Já as Figuras 7 e 8 apresentam, respectivamente,

	Modularidade	Tempo(s)	N° de Comunidades
Betwenness	0,9453312	24,440	406
Caminhada Aleatória	*	*	*
FastGreedy	0,8252987	0.014	396
Autovetores	0,81933	48,564	234
Spinglass	*	*	*

Tabela 4. Rede *Netscience*.

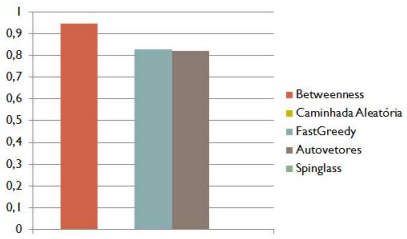

Figura 4. Modularidade obtida na rede *Netscience* pelos diferentes algoritmos de detecção.

a relação entre tempo de execução versus quantidade de vértices e tempo de execução versus quantidade de arestas.

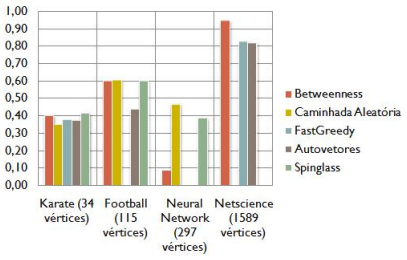

Figura 5. Relação entre medida de modularidade versus quantidade de vértices.

Observando os gráficos contidos nas Figuras 5 e 6 percebe-se que com poucos vértices e arestas, os diferentes algoritmos apresentam valores de modularidade similares. No entanto, com o aumento do número de vértices e arestas, não é possível estabelecer um padrão. Em relação aos gráficos das Figuras 7 e 8 nota-se que o tempo de execução aumenta quando o número de vértices e arestas também aumentam, exceto para os algoritmos *Caminhada Aleatória* e *FastGreedy*. Nestes dois casos, mesmo para um grande número de vértices e arestas, o tempo de execução permanece baixo. Ressalta-se que o algoritmo *Autovetores* apresentou o maior tempo de execução.

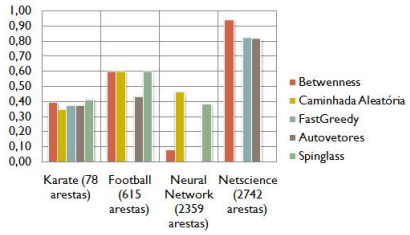

Figura 6. Relação entre medida de modularidade versus quantidade de arestas.

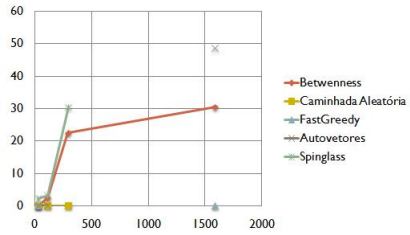

Figura 7. Relação entre tempo de execução versus quantidade de vértices.

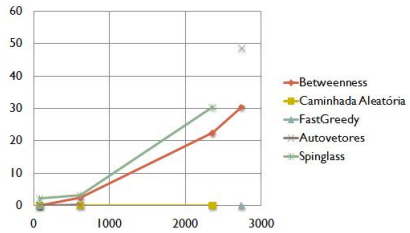

Figura 8. Relação entre tempo de execução versus quantidade de arestas.

5. Conclusão

Neste trabalho foram realizados experimentos em quatro redes conhecidas: *Karate*, *Football*, *Neural Network* e *Netscience*. Tais experimentos tinham por objetivo avaliar algoritmos de detecção de comunidades (*Betwenness*, *Caminhada Aleatória*, *FastGreedy*, *Autovetores* e *Spinglass*), pertencentes à diferentes abordagens, por meio do cálculo da modularidade da partição obtida pelos algoritmos, o tempo de execução e a quantidade de comunidades encontrada na partição de maior valor de modularidade.

Na maioria dos casos, os algoritmos resultaram em partições com altos valores de modularidade. Porém, notou-se que não existe um consenso entre os diferentes algoritmos quando se trata do número de comunidades contidas na partição com maior valor de modularidade. Como tal informação geralmente não é conhecida *a priori*, espera-se que as partições com maiores valores de modularidade resultem na melhor divisão da rede em comunidades.

Os algoritmos apresentam um aumento do tempo de execução à medida que aumenta o número de vértices e arestas da rede. Este padrão não se confirma apenas para os algoritmos *Caminhada Aleatória* e *FastGreedy*, os quais não são influenciados pela quantidade de vértices e arestas, sendo executados rapidamente. Ressalta-se que tais algoritmos também apresentam valores satisfatórios de modularidade.

Ainda faz-se necessário a avaliação dos algoritmos considerando outras bases de dados (inclusive bases com uma maior quantidade de vértices e arestas como, por exemplo, a base Internet) e a comparação com resultados encontrados na literatura. Dessa forma, é possível confirmar mais precisamente o comportamento obtido pelos algoritmos nos experimentos realizados.

Referências

[1] J. P. Bagrow and E. M. Bollt. A local method for detecting communities. *Physical Review E*, 72, 2005.

[2] S. Boettcher. Extremal optimization for graph partitioning. *Physical Review E*, 64(026114), 2001.

[3] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(026132), 2005.

[4] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 79(066111), 2004.

[5] J. Duch and A. Arenas. Community detection complex networks using extremal optimization. *Physical Review E*, 72, 2005.

[6] M. Girvan and M. E. J. Newman. Community estructure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99(12):7821–7826, 2002.

[7] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Science USA*, 99(12):7821–7826, 2002.

[8] igraph. Igraph. http://igraph.sourceforge.net/.

[9] C. P. Massen and J. P. K. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71(046101), 2005.

[10] S. Muff, F. Rao, and A. Caflisch. Local modularity measure for network clusterizations. *Physical Review E*, 72(056107), 2005.

[11] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(066133), 2004.

[12] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physics/0605087*, 74, 2006.

[13] M. E. J. Newman. Modularity and community structure in networks. *National Academy of Sciences of the USA*, 103(23):8577–8582, 2006.

[14] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(026113), 2004.

[15] P. Pons and M. Latapy. Computing communities in large networks using random walks. *Proceedings of the 20th International Symnposium on Computer and Information Sciences, ISCIS'05, LNCS 3733*, 2005.

[16] M. G. Quiles, L. Zhao, R. L. Alonso, and R. A. F. Romero. Particle competition for complex network community detection. *Chaos (Woodbury)*, 18(033107):1–10, 2008.

[17] R. R project. http://www.r-project.org/.

[18] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *cond-mat/0603718*, 2006.

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

[20] M. Tasgin and H. Bingol. Community detection in complex networks using genetic algorithms. *cond-mat/0604419*, 2006.

[21] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

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