XXXIII Congresso de Iniciação Científica da Unicamp

Comparação de métodos de detecção de comunidades em redes complexas

Palavras-Chave: grafos, modularidade, blocos estocásticos

Autores(as):
Nicolas Firmiano Alves, IFGW – UNICAMP
Prof. Dr. José Antônio Brum (orientador(a)), IFGW - UNICAMP

INTRODUÇÃO:

O objetivo principal do trabalho é compreender as diferenças entre as diversas classes de métodos de detecção de comunidades em redes complexas. Ou seja, entender as diferentes maneiras de detecção de sub grafos densamente conectados entre si em uma rede complexa. Esse tópico de estudo se encontra em diversas áreas de conhecimento e atualmente é útil em modelos de inteligência artificial e neurociência. Em particular, serão utilizados os modelos baseados em Modularidade e Blocos Estocásticos, estudando sua base teórica, desenvolvendo programas para simular os algoritmos, tanto em redes reais quanto em redes sintéticas.

A abordagem das duas classes de modelos no problema é diferente, levando a comunidades sutilmente distintas. Os principais algoritmos que usam a modularidade como função qualidade são os algoritmos de Louvain[1] e Leiden[2], em que o algoritmo de Leiden conserta casos específicos onde o Louvain é limitado. Os algoritmos que maximizam a modularidade têm a característica de agrupar nós com alta densidade de links entre si, pela própria maneira que a modularidade é construída, essa não é a mesma lógica dos modelos baseados em blocos estocásticos, (que daqui para frente serão referidos como SBM).

O modelo de SBM que será tratado no trabalho é o Degree Corrected SBM[3], ou DCSBM[4]. Dessa maneira, assim como a maioria dos modelos que usam SBM, o DCSBM é um modelo estatístico que maximiza não mais a densidade interna de links, mas sim encontra a partição que maximiza a probabilidade da rede ser gerada por blocos (comunidades), controlada por parâmetros, mas levando em consideração o grau de cada nó.

METODOLOGIA:

Para mérito de comparação dos dois modelos as redes que usaremos como teste dos algoritmos serão, SBM simétrico, ou seja é uma rede SBM com matriz de probabilidades uniforme e igual na diagonal, o mesmo vale para os termos não diagonais, note que os termos diagonais não necessariamente são iguais aos não diagonais, a vantagem de usar esse tipo de rede é que

manipulamos tanto o número de comunidades quanto a densidade de conexões, escolheremos seus parâmetros de modo a obter uma rede esparsa. Os algoritmos também serão testados nas redes C. Elegans[5], muito usada na neurociência e a rede Karatê[6] muito popular no estudo de comunidades. Também será usada a rede sintética Barabasi-Albert[7]. O objetivo é, entender o papel da resolução, ou o fator γ, na qualidade das partições que maximizam a modularidade, comparar os algoritmos e explicitar o fundamento teórico dos dois modelos.

- **ALGORITMOS BASEADOS NA MODULARIDADE**

A função qualidade do algoritmo de Louvain[1] e Leiden[2] é a modularidade. Ela pode ser definida como a diferença normalizada entre o número de links total de um certo conjunto de nós $\mathrm{C_n}$ e seu valor esperado caso a rede fosse aleatória. Em outras palavras, caso a rede tivesse links igualmente distribuídos entre todos os nós, não existiriam comunidades na rede, pois seria impossível a formação de uma preferência de conexão. Portanto, a modularidade quantifica o nível do afastamento de um conjunto de um regime aleatório. Podemos escrever isso matematicamente como $\mathrm{M_c} = \frac{1}{2\mathrm{L}}\sum_{(\mathrm{i,j})\in \mathrm{C_n}}(\mathrm{A_{ij}} - \mathrm{p_{ij}})$. Dessa forma, o processo do algoritmo Louvain para maximização da modularidade se dá por:

1. Encontrar a partição que maximize a modularidade
   1.1. Considere inicialmente cada nó como sendo uma comunidade
   1.2. . Mova cada nó para a comunidade de seus vizinhos e calcule a variação da modularidade em cada uma delas. Mova o nó para a comunidade com maior ganho positivo de modularidade
   1.3. Repita o passo 1.2 até que a modularidade total não aumente mais
2. Construir uma nova rede cujos nós são dados pelas comunidades encontradas na fase 1, o peso do link entre os nós é a soma dos pesos dos links entre as comunidades e a soma dos pesos dos links das mesmas comunidades se tornam pesos de auto links dos novos nós.
3. Repetir o primeiro e o segundo passo até que a modularidade global seja maximizada

Vale comentar que a maneira como é calculada a variação da modularidade é otimizada e vale $\Delta M = \frac{L_{i\to A}}{L} - \gamma \cdot \frac{k_A k_i}{2L^2}$, Onde L é a quantidade de links total da rede, k_A é a soma do grau de todos os nós dentro da comunidade A, k_i é o grau do nó i e $L_{i\to A}$ é a quantidade de links que um nó i tem com a comunidade A. Já γ é dito como um fator de resolução dado como parâmetro de entrada do algoritmo, em geral, é igual a um, mas quando modificado, é diretamente proporcional ao número de comunidades encontradas. Essa resolução pode compensar a

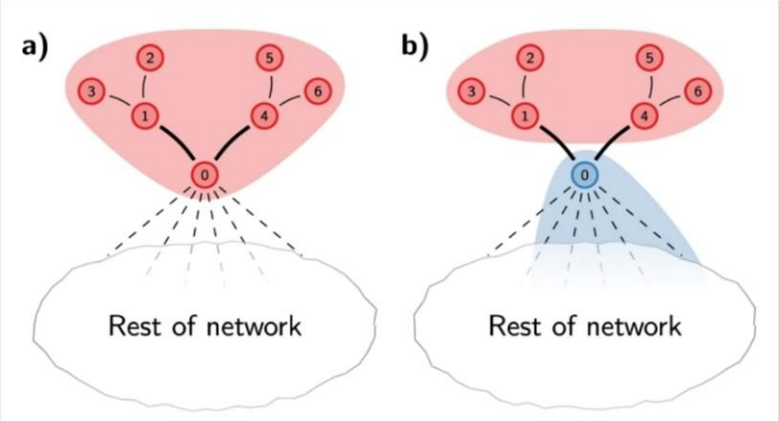

Figura 1 – Exemplo de erro do algoritmo Louvain –Imagem obtida da referência [2]

tendência de algoritmos que usam modularidade de englobar comunidades menores, ao passo que o número de links fica muito grande.

O algoritmo Louvain, falha em certas situações e pode acabar encontrando uma partição desconexa, tome por exemplo o caso da figura 1, um nó está conectando dois grupos dentro de uma mesma comunidade, mas o mesmo possui muitos outros links com o resto da rede. Então, ao tentar maximizar localmente a modularidade, o nó que está exercendo o papel de ponte dentro da comunidade acaba sendo movido para o grupo de um de seus vizinhos, gerando partições ruins. Esse problema é corrigido pelo algoritmo Leiden, que introduz uma fase de refinamento das partições, podemos então formalizar o algoritmo da seguinte forma:

1. Encontrar a partição que maximize a modularidade, assim como no algoritmo Louvain.
2. Ajustar cada uma das partições, ou seja, maximizar a modularidade dentro de cada partição encontrada transformando as comunidades em sub comunidades
3. Construir uma nova rede cujos nós são dados pelas comunidades ajustadas dentro das comunidades originais encontradas na fase 1, o peso do link entre os novos nós é a soma dos pesos dos links entre as sub comunidades e a soma dos pesos dos links das mesmas sub comunidades se tornam pesos de auto links dos novos nós. As comunidades do passo 1 viram comunidades iniciais dos super nós.
4. Repetir o primeiro e o segundo passo até que a modularidade global seja maximizada.

Apesar de introduzir um passo a mais com o processo de refinamento, o custo computacional diminui drasticamente em comparação ao Louvain e principalmente, encontra comunidades sempre conexas.

- **ALGORITMOS BASEADOS NO MODELO DE BLOCOS ESTOCÁSTICOS**

O DCSBM se baseia no modelo de blocos estocásticos na geração de modelos aleatórios, introduzindo em seus parâmetros o grau esperado de cada nó e tem como função qualidade a probabilidade, ou mais precisamente o logaritmo da probabilidade, do modelo aleatório gerar o grafo original. O valor esperado do grau do nó é uma alternativa para contornar o fato que cada comunidade continua sendo gerada por um modelo Erdos-Renyi[8] internamente, levando a comunidades enviesadas que não respeitam a topologia da rede. O SBM possibilita gerar grafos aleatórios mais parecidos com a realidade, dividindo a rede em q grupos, associando a cada grupo uma probabilidade de conexão e, então, conectando a rede. A probabilidade de um nó em uma dada comunidade r ter um link com outro nó de uma comunidade s é P_{rs}, representado por uma matriz qxq. O valor de P_{rs} deve ser restrito a aqueles que preservam o grau esperado dos nós para que não hajam distorções no modelo estatístico, ou seja, se K_i for o grau do nó i e g_i for a comunidade do nó i então $\sum_j p_{r,g_j} k_j = 2m$, com m sendo o número de links total da rede.

A probabilidade do modelo DCSBM gerar o grafo real é uma distribuição de Bernoulli, mas que pode ser aproximada por uma distribuição de Poisson. Por facilidade podemos calcular o logaritmo dessa probabilidade e depois de um extenso processo algébrico expressar essa probabilidade em função apenas da partição dos nós, ou seja, encontrar os três parâmetros (partição dos nós, probabilidades P_{rs} e grau esperado de cada nó) que mais se aproximem do grafo real, se transforma em um problema de otimização de uma função qualidade, conhecida como perfil log-verossimilhança dada por $\mathcal{L} = \sum_{rs} m_{rs}\, ln\left(\frac{m_{rs}}{K_r K_s}\right)$, onde m_{rs} é a quantidade de links entre as comunidades r e s e K_r é soma de todos os graus da comunidade r.

O processo de maximização de $\mathcal{L}$ utilizado no trabalho é o feito por Mark e Newman[4] e tem como valor de entrada o número de comunidades e um número k de vezes que o algoritmo deve ser executado. O algoritmo se dá da seguinte maneira:

1. Formar uma partição inicial aleatória que respeite a quantidade de comunidades de entrada
2. Dentre todos os possíveis movimentos de nós mova o de maior $\mathcal{L}$ total. Cada nó pode ser movido apenas uma vez e todos os nós devem ser movidos
   2.1. Escolha a partição que tenha o maior $\mathcal{L}$ global
3. Repita o passo 2, k vezes ou até haver um decréscimo da função qualidade total

Mais uma vez podemos encontrar uma fórmula para a variação de $\mathcal{L}$, tornando o algoritmo mais eficiente.

RESULTADOS E DISCUSSÃO:

Primeiramente, vamos entender se ambos algoritmos conseguem detectar uma mudança na topologia de uma rede real, para isso, vamos usar o modelo nulo na rede C. Elegans, a rede Barabasi com densidade de links similares e com isso analisar como a função qualidade total muda ao embaralhar a topologia da rede. No caso do algoritmo Louvain vamos variar a resolução e no DCSBM variamos o número de comunidades.

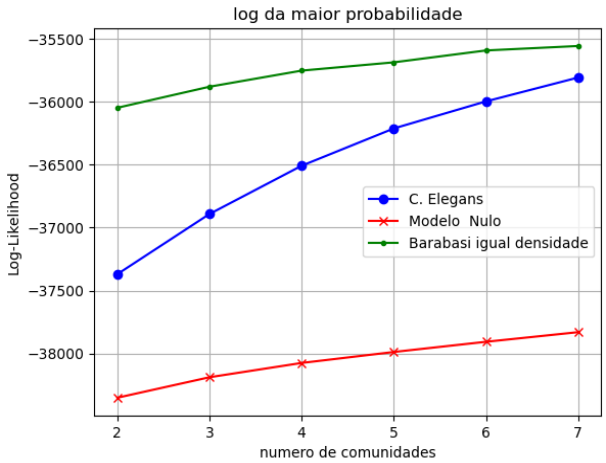

Figura 2 - variação da função qualidade total no DCSBM

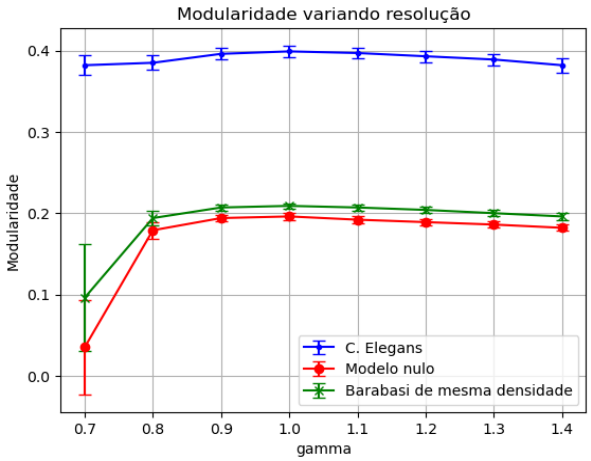

Figura 3 – variação da função qualidade total no Louvain

O DCSBM é mais delicado para fazer comparações de resultados entre redes, já que sua função qualidade não é normalizada e inclusive é monotonicamente crescente conforme o aumento do número de comunidades, o algoritmo Louvain não sofre do mesmo problema. Entretanto, ambos apresentam um aumento substancial da função qualidade em comparação com o modelo nulo.

Modularidade e DCSBM são eficientes e encontram boas partições em tempos razoáveis, embora o DCSBM seja consideravelmente mais lento. Os resultados obtidos do DCSBM possuem base estatística robusta já os baseados na modularidade têm natureza heurística e muitas vezes não garantem um máximo global. Parte importante do funcionamento do DCSBM é assumir que a rede em questão é formada por blocos de conexões parecidas entre si, mas redes aleatórias e muitas redes livres de escala não possuem blocos pois seguem outra lógica de conexão o DCSBM será impreciso nesse limite.

CONCLUSÕES:

Algoritmos baseados em modularidade, tendem a agrupar pequenas comunidades dentro de outras maiores, mesmo quando as mesmas são bem definidas e, em redes muito heterogêneas podem encontrar partições imprecisas, tendo dificuldade em identificar nós com muitos links na mesma comunidade que nós de grau baixo. Já o DCSBM não possui dificuldades em redes heterogêneas pois corrige os graus dos nós em seu modelo, por outro lado, não consegue estipular o número ideal de comunidades e não possui mecanismos para comparar sua função qualidade entre números de comunidades diferentes. Atualmente, estamos aprofundando o estudo do modelo DCSBM e avaliando a melhor maneira de otimizar sua função qualidade e de estipular um número de comunidades ideal.

BIBLIOGRAFIA

[1] V. D. Blondel et al. “Fast unfolding of communities in large networks”. Em: J. Stat. Mech. Theory Exp. 6 (2008), P10008. DOI: doi:10.1088/1742-5468/2008/10/P1000

[2] https://www.nature.com/articles/s41598-019-41695-z

[3] *Holland, Paul W; Laskey, Kathryn Blackmond; Leinhardt, Samuel (1983). "Stochastic blockmodels: First steps". Social Networks. **5** (2): 109–137. doi:10.1016/0378-8733(83)90021-7. ISSN 0378-8733. S2CID 34098453. Archived from the original on 2023-02-04. Retrieved 2021-06-16.*

[4] Stochastic blockmodels and community structure in networks Brian Karrer1 and M. E. J. Newman1

[5] Daniel Witvliet et al. “Connectomes across development reveal principles ofbrain maturation in C. elegans”. Em: Nature 596 (2021), pp. 257–261.

[6] Zachary, W. W. (1977). «An Information Flow Model for Conflict and Fission in Small Groups». Journal of Anthropological Research. 33 (4): 452–473.

[7] Barabási, A.L.; H. Jeonga, Z. Nédaa, E. Ravasza, A. Schubertd, T. Vicsek. «Evolution of the social network of scientific collaborations». *Physica A*