



Detecção de Comunidades em Grafos Egocêntricos: Uma análise comparativa

DOI: 10.37702/2175-957X.COBENGE.2024.5014

Autores: VINICIUS LIMA FERNANDES, GABRIEL RIBEIRO PASSOS, LUIS AUGUSTO MATTOS MENDES, MATHEUS AVILA MOREIRA DE PAULA, EDUARDO GABRIEL REIS MIRANDA

Resumo: Por meio de uma análise comparativa entre os resultados dos algoritmos de detecção de comunidades: Louvain, Leiden, rb pots, surprise communities e threshold communities, foi possível observar discrepâncias significativas na formação das comunidades. Além disso, a pesquisa enfatizou a importância da coerência e qualidade das comunidades detectadas. Os resultados obtidos contribuem para uma melhor compreensão da detecção de comunidades em grafos egocêntricos e ressaltam a relevância da análise comparativa na avaliação da eficácia dos algoritmos empregados.

Palavras-chave: Grafos, Rede egocêntrica, Detecção de comunidades.

DETECÇÃO DE COMUNIDADES EM GRAFOS EGOCÊNTRICOS: UMA ANÁLISE COMPARATIVA

1 INTRODUÇÃO

A utilização da modelagem para resolução de problemas, é uma alternativa para uma adequação acerca das tomadas de decisão, onde é possível olhar para o problema e prever suas dificuldades ou pontos a serem explorados (CORREA; DAVIS JR, 2008, p. 8-11). Dessa forma, a representação de problemas através dos grafos, se apresenta como uma importante ferramenta de modelagem a ser utilizada, para facilitar a visualização e compreensão humana (DE MEDEIROS; DE MELO SILVA; DE ARIMATÉA ROCHA).

Como exposto por Uchoa (2012), a aplicação da modelagem de problemas utilizando grafos, pode ser usada em diversos cenários e áreas, como exemplo, problemas logísticos onde é necessário encontrar a menor rota entre dois pontos distintos. Diante desse contexto, os grafos são uma importante ferramenta utilizada no ambiente educacional e de análise estatística por engenheiros, uma vez que possibilita realizar a modelagem de problemas e otimizar as soluções (FERREIRA, 2014).

Com isso, apesar de existir diversos tipos de grafos, encontram-se muitas aplicações de modelagem que utilizam grafos egocêntricos. Os grafos egocêntricos são um tipo especial de grafo, onde o vértice central (ego) está conectado a todos os demais vértices (alters). Desse modo, a análise do grafo egocêntrico se baseia a partir das relações entre os alters com o ego central (GARTON; HAYTHORNTHWAITE; WELLMAN, 1997).

Como apresentado por Bogado (2022), através da utilização de grafos egocêntricos, foi possível realizar a análise de coautorias em publicações científicas entre Brasil e demais países a respeito da Hanseníase, onde o Brasil foi definido como vértice ego e os demais países como alters, possibilitando assim, a utilização de grafos como uma ferramenta para a análise estatística na área educacional.

Além disso, a evolução tecnológica, a indústria e o mercado de trabalho estão demandando, dos estudantes da área de ciências exatas e computação, a aquisição de conhecimentos teóricos e o desenvolvimento de habilidades práticas convergentes com os desafios da indústria 4.0. Portanto, para enfrentar os desafios tecnológicos impostos pelo avanço da tecnologia é imprescindível o aprimoramento do processo de ensino-aprendizagem de forma a integrar concretamente teoria e prática. (GLOVER *et al.*, 2016; ROY, 2022).

Por conseguinte, a obtenção de conhecimentos teóricos aliado ao desenvolvimento de habilidades práticas no uso de técnicas de análise de grafos habilita os estudantes a propor soluções nas mais variadas áreas como transporte, energia, bioinformática, redes de comunicação, qualificando-os para os desafios do mercado de trabalho.

Dessa forma, o objetivo central do presente estudo, é realizar a análise visual das discrepâncias entre comunidades de grafos egocêntricos formadas por diferentes algoritmos, além da análise comparativa da similaridade entre as comunidades esperadas e as geradas pelos algoritmos. Este estudo também objetiva abordar a análise de grafos, como uma área da mineração de dados utilizando, para isso, a linguagem de programação Python como ferramenta auxiliar para o ensino e aprendizado de análise de algoritmos e grafos.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 é apresentado o Referencial Teórico; na Seção 3 são apresentados os Materiais e Métodos utilizados; a Seção 4 apresenta a Discussão e Resultados; e, por fim, na Seção 5 são apresentados as Considerações Finais.

2 REFERENCIAL TEÓRICO

A respeito da utilização dos grafos, uma das aplicações que podem ser realizadas, trata da detecção e análise sobre comunidades. Ao representar uma rede de relações entre pessoas, utilizando grafos, cada vértice representa um indivíduo e cada aresta representa o tipo de relação existente entre dois indivíduos. Uma comunidade, pode ser definida como um agrupamento de indivíduos que possuem características semelhantes, que pode ser extraída do cenário apresentado anteriormente, onde os conjuntos de vértices são construídos através das semelhanças entre suas relações. Dessa forma, a principal determinação que movimenta a busca por detecção de comunidades é a necessidade de identificar agrupamentos diferentes possibilitando compreender as principais características de cada comunidade encontrada (MACHADO, 2011).

Portanto, diversas medidas para a detecção de comunidades são utilizadas, dentre as quais se encontra a modularidade, que foi desenvolvida e aprimorada por Newman. Através dela é possível avaliar a qualidade das comunidades formadas permitindo uma melhor avaliação a respeito das diferenças. Deste modo, quanto maior a modularidade, melhor definida será uma comunidade e maior será a distinção entre comunidades diferentes (BEDI; SHARMA, 2016).

Para a medida de modularidade podemos considerar os algoritmos de Louvain, Leiden, Randomized Binary Partitioning on Trees for Community Detection (RB Pots). Como alternativa a modularidade existe a medida de surpresa representado pelo algoritmo Surprise Communities, outra opção seria o algoritmo Threshold Communities que utiliza um valor limiar para realizar a separação das comunidades. Todas as definições dos algoritmos estão disponíveis na documentação da biblioteca CDlib¹.

Seguindo o princípio da maximização da modularidade, foi projetado o algoritmo de Louvain, que é utilizado para identificação de comunidades em grafos, por possuir um tempo de execução linear. Para realizar a detecção das comunidades, o algoritmo adota dois passos específicos. No primeiro passo todos os nós são colocados em comunidades diferentes; no segundo passo ocorre a movimentação entre os nós da comunidade buscando o ganho de modularidade. Quando esse ganho é positivo o nó é transferido da comunidade de origem para a de destino (BLONDEL, 2008).

No entanto, o algoritmo de Louvain apresenta como deficiência a impossibilidade de garantir a conectividade das comunidades. Então, por mais que seja identificado o grupo de nós que estão mais densamente conectados não se tem a garantia de que todos eles estejam diretamente conectados entre si. Nesse sentido, o algoritmo de Leiden foi elaborado seguindo o mesmo princípio visando corrigir a deficiência apresentada referente ao algoritmo de Louvain (RODRÍGUEZ RODRÍGUEZ; RODRÍGUEZ; ELIOZONDO MORENO, 2022).

Outro algoritmo, que também busca a maximização da modularidade é denominado Randomized Binary Partitioning on Trees for Community Detection (RB Pots). Sua ideia consiste basicamente em particionar o grafo em subárvores de forma aleatória e binária. Após isso, utiliza-se da modularidade para melhorar a qualidade de cada partição encontrada anteriormente (REICHARDT; BORNHOLDT, 2006).

¹ https://cdlib.readthedocs.io/en/latest/reference/cd_algorithms/node_clustering.html

Surprise Algorithm for Community Detection (Surprise communities) é um algoritmo que utiliza uma medida chamada surprise (surpresa), que é baseada na probabilidade clássica, de forma que busca maximizar essa medida. Esse algoritmo não se concentra unicamente na densidade de conexões dos nós, mas também leva em consideração o número esperado de conexões dos nós e compara com o número real de conexões existentes (TRAAG; ALDECOA; DELVENNE, 2015).

O algoritmo Limiar de Similaridade Semântica (Threshold communities) é baseado no critério de similaridade e conexão, o agrupamento ou não de um nó em uma comunidade é baseado no valor limiar escolhido. Dessa forma, conforme o valor limiar escolhido o algoritmo decide se dois vértices são similares o suficiente para serem agrupados (GUZZI; VELTRI; CANNATARO, 2014).

3 MATERIAIS E MÉTODOS

Visa a comparação entre as comunidades geradas pelos algoritmos e a comparação com a comunidade esperada foram definidas 5 etapas: ETAPA 1: Elaborar uma lista de pessoas (alters) com quem o vértice (ego) possui algum tipo de relação; ETAPA 2: Classificar as relações existentes entre os vértices; ETAPA 3: Definir as comunidades esperadas separando as pessoas em grupos com características em comum; ETAPA 4: Aplicar os algoritmos para detecção das comunidades; ETAPA 5: Comparar os resultados obtidos.

O desenvolvimento do projeto envolve a compreensão do contexto no qual o projeto está inserido e o comportamento dos algoritmos selecionados. Após esse entendimento deu-se início a ETAPA 1. Para a execução desta etapa e buscando automatizar a construção da lista de pessoas foi construído um script utilizando a linguagem Python². A tela utilizada para o Cadastro de Pessoas é apresentada na Figura 1.

Figura 1 – Tela para Cadastro de Pessoas.

Fonte: autor

A tela de Cadastro de Pessoas possui os seguintes campos: Arquivo de nomes, Nome, Apelido, Cidade. O campo Arquivo de nomes refere ao arquivo em que foram inseridas, uma a uma, as informações sobre as pessoas que compõem a rede de conexões. Para cada pessoa que foi incluída na ego rede, foi informado o nome, apelido (opcional) e a cidade em que a pessoa reside. Uma vez realizado o cadastro de todas as

² <https://www.python.org/>

peças, passou-se para a ETAPA 2 utilizando-se um novo script desenvolvido na linguagem Python para agilizar o processo de classificação das relações.

A Figura 2 representa o script em execução:

Figura 2 – Tela para Classificação das Relações.

Fonte: autor

Nesta etapa, foi gerado um arquivo contendo as conexões entre as pessoas inseridos no arquivo da ETAPA 1. A tela de Classificação das Relações possui os seguintes campos: Arquivo pessoas, Arquivo conexões, Origem(ns) e Destino(s). O campo Arquivo pessoas refere-se ao arquivo construído na ETAPA 1. O campo Arquivo conexões refere ao arquivo em que foram inseridas, uma a uma, o tipo de relação entre as pessoas selecionadas. Para a seleção dessas pessoas foram utilizados os campos Origem(ns) e Destino(s), onde clicando no botão '+' ao lado do campo 'Origem(ns)', essa ação abre uma nova janela, representada pela Figura 3, onde mostra as pessoas que foram cadastradas na ETAPA 1. Selecionadas as pessoas de origem, o mesmo processo é repetido para o campo Destino(s).

Figura 3 – Janela de seleção de pessoas.

Fonte: autor

Como último passo da ETAPA 2, foi feita a seleção do tipo de relação, que foi salva no arquivo de texto inserido no campo 'Arquivo conexões'. São 13 as opções de classificação apresentadas no Quadro 1:

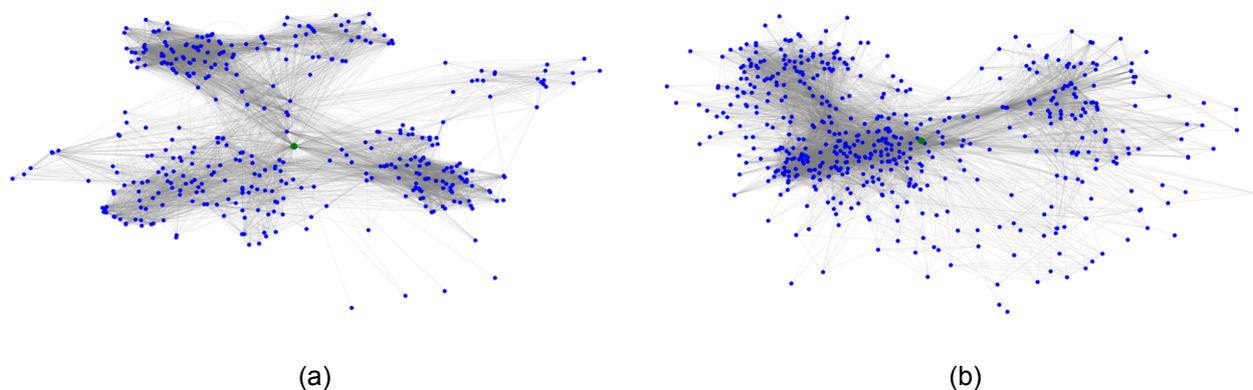
Quadro 1 - Tipos de relações.

Peso	Relação
1	Família
2	Melhor Amigo(a)
3	Cônjuge/Namorado(a)
4	Amigo
5	Parente
6	Parça
7	Professor
8	Aluno
9	Colega
10	Trabalho
11	Consanguíneo
12	Ficante
13	Conhecido

Fonte: autor

As relações inseridas no arquivo estão representadas individualmente, da seguinte forma: "ID da pessoa de origem - ID da pessoa de destino - Tipo da relação". A Figura 4 representa as redes ego construídas a partir dos dados obtidos nas etapas anteriores, e sua visualização foi construída na linguagem Python com o auxílio das bibliotecas NetworkX³ e Matplotlib⁴:

Figura 4 – Redes ego - (a) base de dados 1; (b) base de dados 2.



Fonte: autor

Na ETAPA 3, foram elaboradas as comunidades esperadas de forma manual pelos autores, ou seja, como era esperado que os algoritmos realizassem a separação das comunidades. Na ETAPA 4, foram aplicados os algoritmos: Louvain, Leiden, RB Pots, Surprise Communities e Threshold Communities, com o auxílio da biblioteca CDlib⁵.

Na ETAPA 5, foram comparados os resultados dos algoritmos aplicados na ETAPA 4 com as comunidades esperadas construídas na ETAPA 3. A comparação foi feita através do algoritmo apresentado na Figura 5.

³ <https://networkx.org/documentation/latest/index.html>

⁴ <https://matplotlib.org/>

⁵ <https://cdlib.readthedocs.io/en/latest/overview.html>

Figura 5 – Pseudocódigo do algoritmo de similaridades de comunidades.

Algorithm 1 Similaridade de Comunidade

Ensure: similaridade(comunidades1, comunidades2, V)

```

1: similares  $\leftarrow 0$ 
2: for all comunidadeX  $\in$  comunidades1 do
3:   maiorSemelhanca  $\leftarrow 0$ 
4:   for all comunidadeY  $\in$  comunidades2 do
5:     interseccao  $\leftarrow$  comunidadeX  $\cap$  comunidadeY
6:     if  $|interseccao| > maiorSemelhanca$  then
7:       maiorSemelhanca  $\leftarrow |interseccao|$ 
8:     end if
9:   end for
10:  similares  $\leftarrow$  maiorSemelhanca
11: end for
12: return similares /  $|V|$ 
    
```

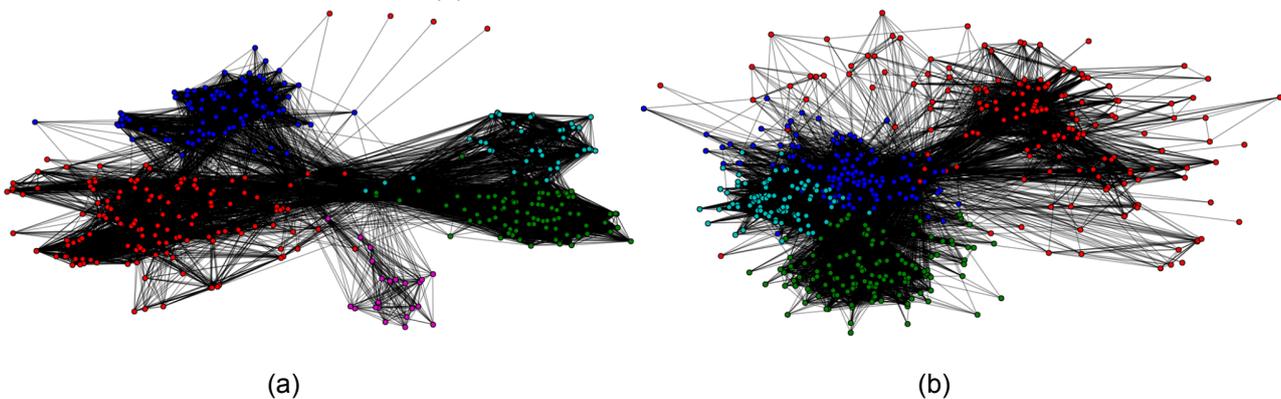
Fonte: autor

Para o funcionamento do algoritmo é necessário a passagem de duas listas de comunidades (X e Y), onde X é a lista de comunidades esperadas e Y é a lista de comunidades geradas a partir da aplicação dos algoritmos para detecção de comunidades. O algoritmo de similaridade busca, para cada comunidade contida em X, uma comunidade em Y que possua a maior similaridade. Esta maior similaridade é representada pela cardinalidade da interseção entre a comunidade N e a comunidade M. O valor de maior similaridade para cada comunidade, é acumulado em uma variável e ao final de todas as buscas, é feita a divisão do valor total pelo número de vértices (pessoas) contidos no grafo. Este valor final, representa a similaridade entre o conjunto de comunidades esperado e o conjunto de comunidades geradas pelos algoritmos.

4 DISCUSSÃO E RESULTADOS

Os resultados obtidos, após a aplicação do algoritmo de detecção de Louvain, podem ser visualizados na Figura 6(a) e Figura 6(b), representando o resultado encontrado de acordo com a base de dados 1 e base de dados 2, respectivamente.

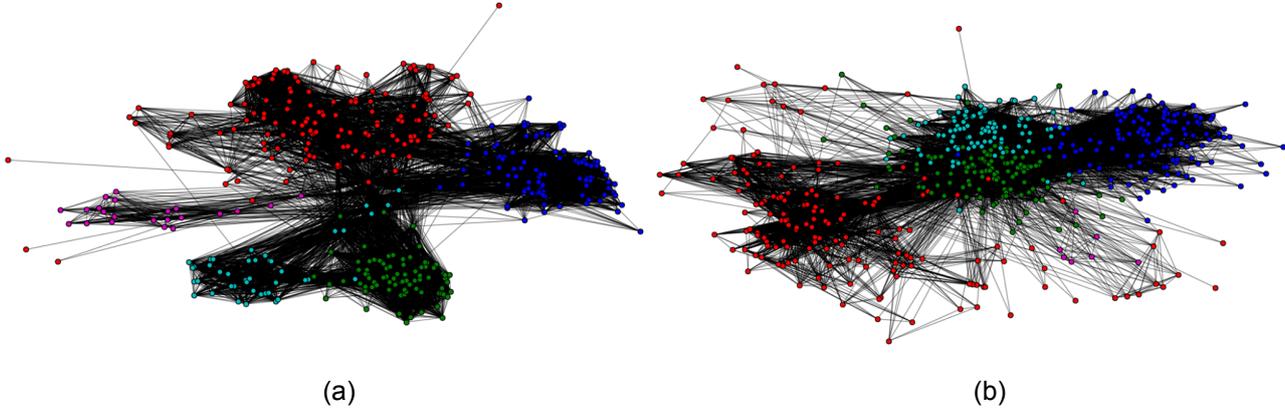
Figura 6 – Comunidades detectadas pelo Algoritmo de Louvain - (a) utilizando a base de dados 1; (b) utilizando a base de dados 2.



Fonte: autor

Os resultados da aplicação do algoritmo de Leiden, para as bases de dados 1 e base de dados 2, podem ser visualizados, respectivamente, na Figura 7(a) e Figura 7(b).

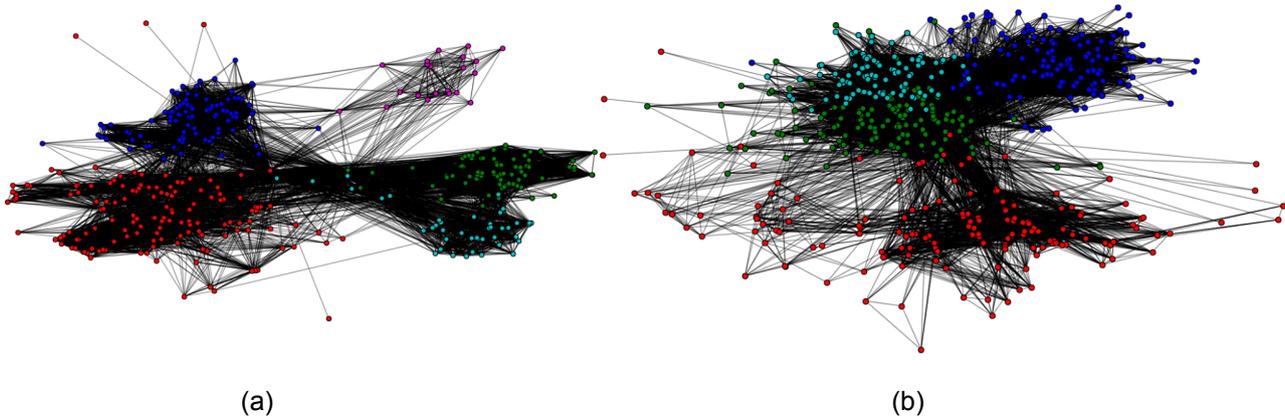
Figura 7 – Comunidades detectadas pelo Algoritmo de Leiden - (a) utilizando base de dados 1; (b) utilizando a base de dados 2.



Fonte: autor

Os resultados da aplicação do algoritmo RB Pots, para as bases de dados 1 e base de dados 2, podem ser visualizados, respectivamente, na Figura 8(a) e Figura 8(b).

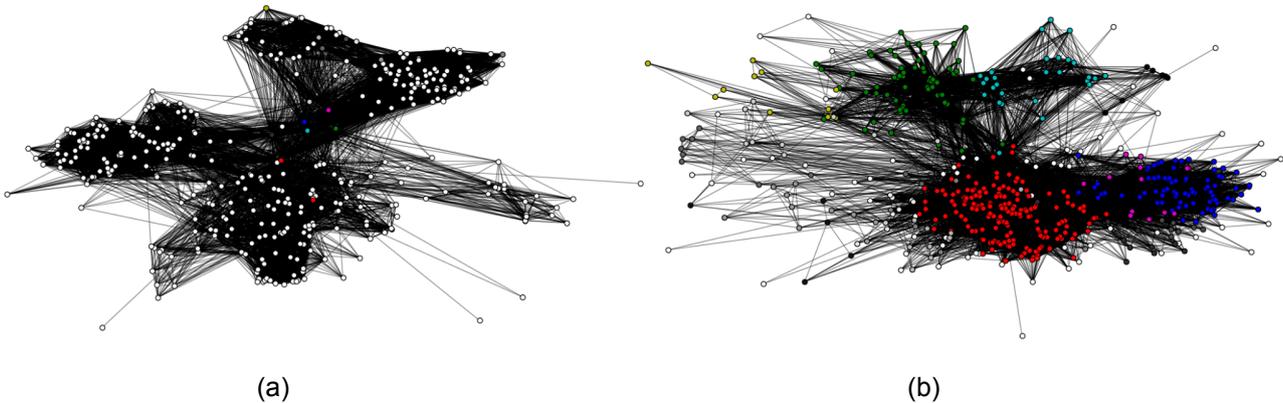
Figura 8 – Comunidades detectadas pelo Algoritmo RB Pots - (a) utilizando base de dados 1; (b) utilizando a base de dados 2.



Fonte: autor

O resultado da aplicação do algoritmo Surprise Communities, para as bases de dados 1 e base de dados 2, podem ser visualizados, respectivamente, na Figura 9(a) e Figura 9(b).

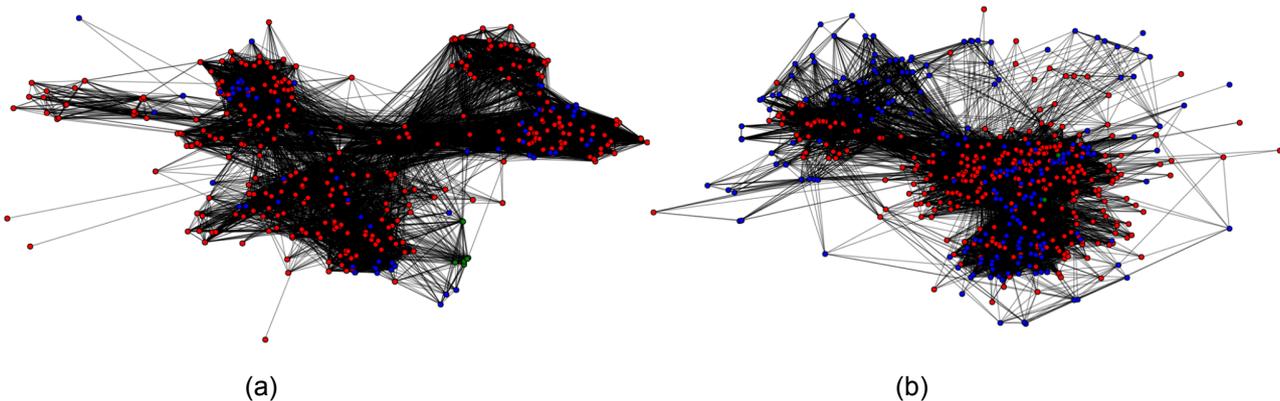
Figura 9 – Comunidades detectadas pelo Algoritmo Surprise Communities - (a) utilizando base de dados 1; (b) utilizando a base de dados 2.



Fonte: autor

O resultado da aplicação do algoritmo Threshold Communities, para as bases de dados 1 e base de dados 2, podem ser visualizados, respectivamente, na Figura 10(a) e Figura 10(b).

Figura 10 – Comunidades detectadas pelo Algoritmo Threshold Communities - (a) utilizando base de dados 1; (b) utilizando a base de dados 2



Fonte: autor

Analisando os resultados encontrados, foi observado uma discrepância visual ao ser comparado o resultado de dois algoritmos diferentes, Louvain e Surprise Communities, acerca do mesmo grafo. A Figura 6(a) ilustra as Comunidades detectadas pelo Algoritmo de Louvain com os dados referentes ao banco de dados 1. Ao analisá-la são identificadas cinco comunidades, sendo elas:

- Em verde - familiares maternos.
- Em azul - familiares paternos.
- Em rosa - vértices relacionados a uma instituição de ensino superior A. Amigos, colegas e professores.
- Em azul - vértices relacionados a uma instituição de ensino superior B. Amigos, colegas e professores.
- Em vermelho - vértices relacionados à cidade natal e convivência durante o ensino fundamental/médio.

Os vértices que não se enquadraram nas quatro primeiras categorias foram alocados pelo algoritmo na comunidade de cor vermelha, que acabou sendo a maior comunidade. Além disso, os vértices que possuem relação somente com o vértice central (nó ego), também foram alocados neste mesmo grupo.

Na Figura 9(a), apesar de uma grande parte dos vértices terem sido representados pela mesma cor, branca, ao executar o algoritmo Surprise Communities, eles não se agruparam na mesma comunidade. O resultado apresentou muitas comunidades individuais, ou seja, comunidade com apenas um vértice.

Ao se analisar os resultados numéricos obtidos ao executar o algoritmo de similaridade de comunidades o Quadro 2 apresenta os resultados obtidos, os valores contidos no quadro variam entre 0 e 1. Através destes dados foi possível observar que os algoritmos de Louvain, Leiden e RB Pots obtiveram os melhores resultados, variando entre eles qual foi o melhor dependendo da base de dados escolhida. Para a base de dados 1, o melhor resultado foi referente ao algoritmo de Leiden, já para a base de dados 2, o melhor resultado veio da aplicação do algoritmo de Louvain. Já os algoritmos Surprise Communities e Threshold Communities apresentaram uma significativa variação ao alternar entre as bases de dados, indicando a necessidade de uma melhora na configuração dos parâmetros ou um melhor tratamento dos dados. Além disso, ambos apresentaram algumas comunidades que não possuíam relação clara entre os vértices que a pertencem.

Quadro 2 - Similaridade entre as comunidades esperadas e comunidades obtidas.

Base de Dados	Louvain	Leiden	RB Pots	Surprise Communities	Threshold Communities
1	0.9362	0.9719	0.9693	0.0127	0.8061
2	0.9739	0.9312	0.9330	0.6263	0.6654

Fonte: autor

5 CONSIDERAÇÕES FINAIS

O presente estudo levantou uma análise comparativa entre as discrepâncias de diferentes algoritmos de detecção de comunidades, entre eles: Louvain, Leiden, RB Pots, Surprise Communities e Threshold Communities. Através da utilização de grafos egocêntricos, foi possível modelar uma rede de pessoas de maneira a estabelecer relações entre os indivíduos.

Atualmente, o desenvolvimento da análise comparativa acerca da detecção das comunidades em grafos egocêntricos, teve todas as 5 etapas, definidas na seção de Materiais e Métodos concluídas. Portanto, é possível afirmar que a discrepância entre o algoritmo que melhor representou o grafo esperado de comunidades e o algoritmo que ficou menos similar a ele, se deve ao fato de como o algoritmo realiza sua análise para agrupar os nós pertencentes ao grafo, de forma que, a influência dos parâmetros utilizados pode ser notada nos resultados obtidos.

Da mesma forma, pode-se destacar que, além de buscar avaliar a qualidade das comunidades geradas pelos algoritmos, também se buscou uma coerência das mesmas, o que através de uma análise empírica, fica evidente que para os algoritmos que apresentaram comunidades menos similares, houve algumas falhas de coerência em

algumas comunidades geradas. Estas falhas são visualizadas pela formação de comunidades que incluem indivíduos sem relação nenhuma com os indivíduos restantes.

AGRADECIMENTOS

Nossos agradecimentos ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e ao Laboratório de Iniciação Científica e Extensão da Computação (LINCE) pelo apoio para a realização deste trabalho.

REFERÊNCIAS

BEDI, Punam; SHARMA, Chhavi. Community detection in social networks. **Wiley interdisciplinary reviews: Data mining and knowledge discovery**, v. 6, n. 3, p. 115-135, 2016.

BLONDEL, Vincent D. et al. Fast unfolding of communities in large networks. **Journal of statistical mechanics: theory and experiment**, v. 2008, n. 10, p. P10008, 2008.

BOGADO, Andréa Cristina; ROSAS, Fábio Sampaio; DE OLIVEIRA, Ely Francina Tannuri. ANÁLISE DAS COAUTORIAS ENTRE BRASIL E DEMAIS PAÍSES EM HANSENÍASE.

CORREA, PRISCILLA OTONI; DAVIS JR, CLODOVEU AUGUSTO. Modelagem de dados geográficos para sistema urbano de tratamento e distribuição de água. **II Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação. Recife-PE**, p. 8-11, 2008.

DE MEDEIROS, Anderson Maíke Nascimento; DE MELO SILVA, Allyson Rafael; DE ARIMATÉA ROCHA, Cristiane. Modelando Problemas Matemáticos por Meio da Teoria dos Grafos. In: V Encontro de Matemática do Agreste Pernambucano, 2018, Caruaru. **Anais**. Caruaru. Disponível em: https://32f4ef4f-8bbc-4a98-92ac-b3f96fdf30bb.filesusr.com/ugd/72f1f0_2c628ff8f6de4ffb93f326d773e6187e.pdf. Acesso em: 03 mai.2024.

FERREIRA, Rafael Agostinho; DE OLIVEIRA, Deive Ciro. Análise Descritiva em Grafos de Produção Científica em Grupo da Universidade Federal de Alfenas. **Cadernos de Estudos Interdisciplinares**, v. 1, n. 1, 2014.

GARTON, Laura; HAYTHORNTHWAITE, Caroline; WELLMAN, Barry. Studying online social networks. **Journal of computer-mediated communication**, v. 3, n. 1, p. JCMC313, 1997.

GLOVER, I., HEPPELSTONE, S., PARKIN, H. J., RODGER, H., IRWIN, B. Pedagogy first: realising technology enhanced learning by focusing on teaching practice. **British Journal**

of **Educational Technology**, v. 47, p. 993-1002, 2016. Disponível em: <https://doi.org/10.1111/bjet.12425>.

GUZZI, Pietro Hiram; VELTRI, Pierangelo; CANNATARO, Mario. Thresholding of semantic similarity networks using a spectral graph-based technique. In: **New Frontiers in Mining Complex Patterns: Second International Workshop, NFMCP 2013, Held in Conjunction with ECML-PKDD 2013, Prague, Czech Republic, September 27, 2013, Revised Selected Papers 2**. Springer International Publishing, 2014. p. 201-213.

MACHADO, Felipe Menezes. **Deteção de comunidades em grafos multicamada muito grandes**.2011. Dissertação (Mestrado) - Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, 2011. Disponível em: <https://repositorio.ufmg.br/handle/1843/SLSS-8LWMYE>. Acesso em:3 mai,2024:

REICHARDT, Jörg; BORNHOLDT, Stefan. Statistical mechanics of community detection. **Physical review E**, v. 74, n. 1, p. 016110, 2006.

RODRÍGUEZ RODRÍGUEZ, I.; RODRÍGUEZ, J. V.; ELIOZONDO MORENO, A. La publicación en Estudios de Género en el último lustro. Un análisis cuantitativo. **iQual**, n. 5, p. 28–50, 2022.

ROY, A. (2022). Diversos Critérios da Tecnopedagogia no Processo Ensino-Aprendizagem. **Revista Internacional de Pesquisa Avançada em Ciência, Comunicação e Tecnologia**. Disponível em: <https://doi.org/10.48175/ijarsct-5683>. Acesso em: 23 mai. 2024.

TRAAG, Vincent A.; ALDECOA, Rodrigo; DELVENNE, J.-C. Detecting communities using asymptotical surprise. **Physical review e**, v. 92, n. 2, p. 022816, 2015.

UCHOA, Joel Silva. **Caminhos mínimos com recursos limitados**.2007. Tese (Doutorado). Curso de Ciência da Computação. Universidade de São Paulo, 2012. Disponível em: https://www.teses.usp.br/teses/disponiveis/45/45134/tde-14012013-165742/publico/dissertacao_joeluchoa_6659698.pdf. Acesso em:3 mai.2024:

DETECTION OF COMMUNITIES IN EGOCENTRIC GRAPHS: A COMPARATIVE ANALYSIS

Abstract:

The present study presents a comparative analysis of community detection using of egocentric graphs, where the central node (ego) is connected to all other vertices (alters). The analysis using graphs was conducted using Python as an auxiliary tool for teaching and learning. Five different community detection algorithms were analyzed: Louvain,

Leiden, RB Pots, Surprise Communities, and Threshold Communities. The analysis was conducted in five stages: construction of the list of people related to the central node, classification of relationships, definition of expected communities, application of the algorithms, and comparison of the obtained results. The results indicated that the Louvain, Leiden, and RB Pots algorithms showed better performance in community detection. On the other hand the Surprise Communities and Threshold Communities algorithms showed significant discrepancies when compared to the expected results.

Keywords: *Graphs, Egocentric Network, Community Detection.*

