

09/10/2020

## Atividade 2: Análise da distância de vértices

Entrega: 22/10/2020 no Portal didático

Atividade individual

---

Tomado um grafo de uma aplicação real<sup>1</sup>, obtenha uma matriz de distâncias<sup>2</sup> e, a partir dela, realize uma análise da posição dos vértices, de maneira semelhante à feita na nossa última aula.

Além desse documento, estou disponibilizando um vídeo no youtube (<https://youtu.be/l9DKG9kHYus>), no qual eu discuto o trabalho.

Obsevações:

<sup>1</sup>Vários repositórios públicos podem ser encontrados na web. Alguns deles são:

- <http://www-personal.umich.edu/~mejn/netdata/>  
Repositório do Mark Newman, um importante pesquisador de comunidades que possui várias bases de escala pequena. Bom para “provar conceito”.
- <http://snap.stanford.edu/data/index.html>  
Repositório do Jure Leskovec (o mesmo da biblioteca). Também é bom para trabalhos sobre comunidades e possui várias bases grandes/gigantes (possui bases com comunidades anotadas). Bom para mostrar aplicações de larga escala.
- <https://deim.urv.cat/~alexandre.arenas/data/welcome.htm>  
Repositório do Alex Arenas, que tem características parecidas com o do Mark Newman.

Porém, aplicações mais interessantes no contexto de uma disciplina de redes podem ser obtidas em outros repositórios ou de outras fontes (frequentemente algum aluno me mostra alguma base de dados legal que encontrou por aí). Uma fonte imensa de dados é o Kaggle (<https://www.kaggle.com/>), que é de onde eu acho que a maioria dos alunos tem tirado bases ultimamente.

<sup>2</sup> As distâncias podem ser calculadas utilizando algoritmos prontos de biblioteca ou qualquer outra fonte, ou seja, **não é necessário implementá-los**. Assim, podem ser utilizadas soluções encontradas na web, desde que o funcionamento seja verificado e a fonte citada. Podem também ser utilizadas bibliotecas de software disponíveis na web com esse propósito. As bibliotecas mais populares são muito fáceis de serem instaladas, principalmente sua versão em python, então qualquer uma delas será bem fácil de usar, das quais vale destacar:

- **igraph** (<https://igraph.org/python/>): Uma das mais bem documentadas e discutidas em fóruns (como stackoverflow), possui uma grande variedade de funções para manipulação e análise básica/clássica. É a que eu estou mais habituado a usar, embora eu saiba de algumas de suas limitações, como o fato de ela não ter atualização recente, não tendo implementados vários algoritmos mais modernos e o fato de ter sua eficiência muito baixa para grafos grandes, principalmente quando se vai buscar vértices específicos, o que é uma operação muito lenta e obriga a utilização de estruturas de dados auxiliares para big data.
- **networkx** (<https://networkx.github.io/>): É também muito bem documentada e também muito discutida em fóruns, com um conjunto muito parecido de funções da igraph, embora ofereça menos algoritmos para análise mais intermediária (como comunidades). Uma das suas vantagens é o fato de que várias bibliotecas auxiliares (como bibliotecas de geoprocessamento e análise de redes urbanas) utilizam a networkx como base, então conhecê-la pode diminuir a curva de aprendizado caso seja importante utilizar alguma solução desse tipo. Também sofre de gargalo de desempenho, de forma semelhante à igraph, quando é necessário percorrer vértices.
- **graph-tool** (<https://graph-tool.skewed.de/>): É uma biblioteca um pouco mais nova, mas muito bem documentada (embora seja menos utilizada em fóruns). É desenvolvida por um brasileiro (Tiago Peixoto), professor da Universidade da Europa Central, que é um importante pesquisador da área de

redes atualmente. Uma das grandes vantagens da graph-tool é a preocupação do desenvolvedor em manter a biblioteca atualizada, sendo capaz de executar grafos grandes, e com algoritmos que incorporam avanços recentes e sua grande atividade em mantê-la/aprimorá-la. Atualmente, é a única biblioteca que eu vejo com atividade constante e acredito que vale muito a pena aprender a utilizá-la (acredito que em breve será a mais utilizada).