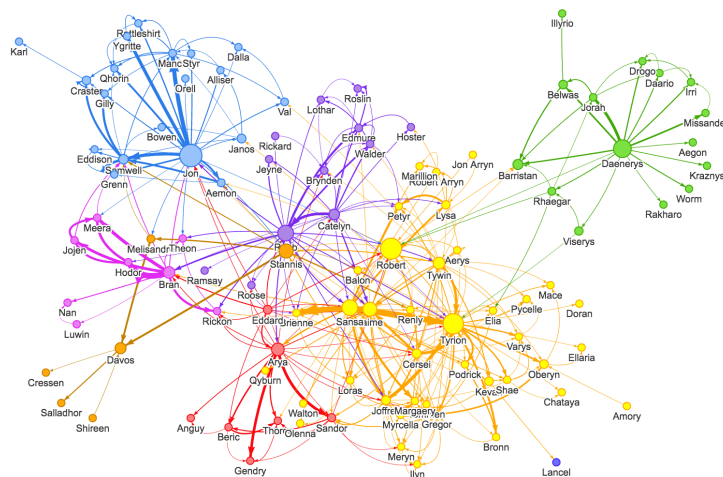


Laboratório ⟨grafos⟩

1 Análise de redes complexas

Grafos são frequentemente usado para modelar e analisar cenários do mundo real, como redes de tráfego [1], dinâmicas de propagação de doenças [2] ou redes de colaboração [3]. Uma aplicação interessante é na análise da interação entre personagens em enredos, como obras literárias, filmes ou séries [4, 5, 6].



As métricas estudadas em sala de aula fornecem ferramentas interessantes para analisar redes de interação, como os graus dos vértices, a densidade do grafo, os tamanhos dos subgrafos completos e o comprimento dos caminhos entre vértices. Além disso, existem outras métricas (algumas específicas para esse tipo de rede) interessantes, como medidas de centralidade (e.g. *betweenness*) e *PageRank*. Também estão disponíveis muitas ferramentas para tratar e analisar grafos. Exemplos incluem NetworkX¹, Gephi², graph-tool³, Neo4J⁴ e igraph⁵. Finalmente, há muito material (principalmente livros e artigos científicos de qualidade) sobre análise de redes sociais. Um bom exemplo é encontrado em <https://ericmjl.github.io/Network-Analysis-Made-Simple>.

Faça um estudo das redes de interação na série *Game of Thrones*. Os dados (personagens e interações) podem ser obtidos em <https://github.com/mathbeveridge/gameofthrones>. Construa os grafos de interações das diferentes temporadas da série e defina o conjunto de métricas (e algoritmos) para analisar essas redes. Além de computar os valores para essas métricas, gere visualizações das redes para sua inspeção visual. Uma sugestão é seguir as abordagens adotadas em outros trabalhos de análise de redes sociais.

Ao final, produza um relatório com os resultados (métricas e visualizações). Discuta esses resultados e apresente as conclusões obtidas. Compare os achados com outros trabalhos da literatura.

2 Maximizando as vendas

A *Tenises* comercializa tênis no atacado. Com grande frequência, a empresa participa de feiras de calçados por todo o país, onde expõe seus produtos a varejistas e fecha contratos para fornecimento de tênis dos

¹<https://networkx.org>

²<https://gephi.org>

³<https://graph-tool.skewed.de>

⁴<https://neo4j.com>

⁵<https://python.igraph.org>

mais diversos modelos. A experiência mostra que quanto maior a variedade de produtos (i.e. tênis com formatos, cores e estilos diversos), maior a probabilidade de venda e sucesso nos contratos. Diante dessa informação, a empresa fez um esforço para classificar cada modelo de tênis conforme uma série de características. Com isso, a empresa consegue determinar, quantitativamente, quão diferentes são dois tênis um do outro. Mesmo com todo esse trabalho realizado, decidir quais modelos levar para a feira é uma tarefa difícil, pois os stands comportam um número limitado de unidades e a empresa possui muitos modelos no seu catálogo. Vamos implementar um algoritmo para decidir quais modelos de tênis levar à feira, de modo que esse grupo seja o mais diverso (i.e. diferente) possível!



Seja n o número de modelos de tênis disponíveis em estoque para serem levados à feira, e m o número de modelos de tênis que serão levados. Um arquivo fornece, para cada par de modelo, a diferença (diversidade) entre eles. O cenário é modelado usando um grafo completo e ponderado, onde cada modelo de tênis é um vértice, e o peso da aresta que conecta um par de vértices define a diferença entre os modelos correspondentes. Esse grafo possui n vértices, e devemos selecionar um subconjunto com m deles, de modo que o somatório de pesos das arestas desse subconjunto seja o maior possível.

Para resolver esse problema, vamos assumir que a solução seja representada por um *array* binário S , onde uma posição i desse *array* indica se o modelo i de tênis será levado (valor 1) ou não (valor 0). Portanto, S tem tamanho n e deve conter m valores iguais a 1, com o restante igual a 0. Um primeiro algoritmo para esse problema constrói a solução passo a passo, selecionando um tênis por vez. O pseudocódigo desse algoritmo é apresentado abaixo.

```
1  CONSTRUCAO():
2    S[i] = 0, para i = {1,2,...,n}
3    PARA i = 1 ATÉ m FAÇA
4      v = seleciona um modelo ainda não escolhido (i.e. S[v] é 0)
5      S[v] = 1
6    RETORNA S
```

O passo da linha 4 é o mais importante desse algoritmo. Temos duas opções: (1) selecionar um modelo ainda não escolhido aleatoriamente (neste caso, construímos uma solução aleatória); ou (2) usar alguma estratégia de seleção mais sofisticada. Como estratégia de seleção, podemos iniciar o algoritmo escolhendo o primeiro modelo aleatoriamente (primeira iteração do laço); nas iterações subsequentes, escolhemos o modelo mais diferente daqueles já escolhidos. Com isso, esperamos construir uma solução melhor.

Um segundo algoritmo para esse problema, baseado em busca local (ou refinamento), consiste em fazermos pequenas modificações na solução, buscando melhorá-la. O pseudocódigo desse algoritmo é apresentado abaixo.

```
1  REFINAMENTO(S):
2    ENQUANTO há melhoria na solução FAÇA
3      M = gera o conjunto de soluções modificadas
4      S = melhor solução de M (que seja melhor que S)
5    RETORNA S
```

Perceba que esse algoritmo recebe como parâmetro uma solução, que pode ser uma solução construída

aleatoriamente ou usando a estratégia de escolha descrita acima. A modificação (usada para gerar o conjunto M da linha 3) consiste em remover um modelo escolhido da solução (atribuindo 0 à sua posição em S) e inserir um modelo não escolhido (atribuindo 1 à sua posição em S). O conjunto M é composto pelas soluções geradas a partir da aplicação dessa modificação a todos os pares de modelos $\{i, j\}$, tal que $S[i] \neq S[j]$.

Implemente esses algoritmos para decidir quais modelos levar à feira. Os arquivos contendo as informações (chamados de instâncias) estão disponíveis em <https://www.uv.es/rmarti/paper/mdp.html>. Use as instâncias *SOM-a* e *SOM-b*. Faça um relatório contendo os resultados da aplicação desses algoritmos na solução de cada instância, e procure responder às seguintes perguntas (lembre-se: a qualidade da solução é dada pelo somatório das diferenças entre todos os pares dos m modelos de tênis escolhidos):

1. Qual algoritmo tem o melhor desempenho?
2. Qual algoritmo executa mais rápido?
3. O refinamento iniciando de uma solução aleatória é melhor ou pior que o mesmo algoritmo iniciando de uma solução construída com a estratégia de escolha predefinida?
4. Quanto o refinamento consegue melhorar na qualidade da solução previamente construída?

Referências

- [1] Manish Joshi and Theyazn Hassn Hadi. A review of network traffic analysis and prediction techniques. *arXiv preprint arXiv:1507.05722*, 2015.
- [2] Carlos Andre Reis Pinheiro, Matthew Galati, Natalia Summerville, and Mark Lambrecht. Using network analysis and machine learning to identify virus spread trends in covid-19. *Big Data Research*, 25:100242, 2021.
- [3] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.
- [4] Vincent Labatut and Xavier Bost. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40, 2019.
- [5] Melody Yu. Decoding the popularity of tv series: A network analysis perspective. *arXiv preprint arXiv:2307.05329*, 2023.
- [6] Ana Lúcia Cetertich Bazzan. Similar yet different: the structure of social networks of characters in seinfeld, friends, how I met your mother, and the big bang theory. *Revista de Informática Teórica e Aplicada*, 27(2):66–80, 2020.